IMPROVED OPTIMALITY CONDITIONS

FOR THE WAGNER-WHITIN ALGORITHM

by

HA YIU CHEUNG, ALBERT

夏耀祥

THESIS

Presented to

The Graduate School

In Partial Fulfilment

of the Requirements for the Degree of

MASTER OF BUSINESS ADMINISTRATION

TWO-YEAR MBA PROGRAMME

THE CHINESE UNIVERSITY OF HONG KONG

May 1988

THESIS COMMITTEE:

Dr. K. H. Lee

Dr. Japhet S. Law

Mr. C. I. Chan

Professor A. M. Chung

TABLE OF CONTENTS

ABSTRACT

The dynamic lot size problem is one of the fundamental areas of research in inventory theory. For the case of uncapacitated, single-item production with no backlogging, this problem can be solved by the Wagner-Whitin algorithm.

In this thesis, a set of optimality conditions known as the network constraints is developed based on a network representation of the dynamic lot size problem, and is found to be valid even when all the costs are time-varying. When these conditions are incorporated into the Wagner-Whitin algorithm to calculate lower bounds for the dynamic program in this algorithm, computation can be reduced. Computation is further reduced by a Lower Bound Theorem derived in this thesis.

Different implementations of the optimality conditions into the algorithm are considered. A simulation experiment is set up to test the performance of these implementations and to assess the impact of several factors on their performance. These factors include the length of planning horizon (n), the time between order value (TBO) and the coefficient of variation of demand (CVD). Results of the experiment are analysed by ANOVA models.

Based on the experimental results, implementation of the optimality conditions significantly improves the computational efficiency of the Wagner-Whitin algorithm. For the original algorithm, only n is found to be a

significant factor. With the implementation of the optimality conditions, significance of the TBO factor increases and the nonlinearity of computational time in n diminishes. In general, CVD is found to be an insignificant factor.

ACKNOWLEDGEMENTS

I wish to express my greatest gratitude to my advisor, Dr. Japhet S. Law, for his encouragement and guidance throughout my graduate studies. His care has extended far beyond this thesis, and his scholarly approach to research has consolidated my determination for an academic career. I look forward to Dr. Law's continuous guidance in the future.

My thanks also go to all other members of the thesis committee : Professor A.M. Chung , Dr. K.H. Lee and Mr. C.I. Chan. Their effort in reviewing this thesis and their invaluable comments are highly appreciated.

I would like to thank Miss Vandy Cheung for her great job in typing. Given the limited time available for completing the work, her patience and enthusiasm are indispensable.

Last, but certainly not least, I must thank my wonderful parents, who have given me the most encouragement, support and love. Without them, none of these would have been possible.

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Research Background

Inventory theory is a discipline which has attracted wide attention from both the academic researchers and the practitioners. For the researchers, the rapid development of this body of knowledge offers ample and challenging research opportunities. For the business and industrial managers, inventory is a significant cost element for production and operations management.

There are three major categories of costs associated with an inventory system [1, 11] :

(a) Holding costs (or carrying costs)

The inventory holding costs are those costs associated with having inventory on hand. They may include the real out-of-pocket costs such as those for storage, handling, insurance and breakage, and the opportunity cost of capital which is incurred by having capital tied up in inventory.

(b) Setup costs (or ordering costs)

These are the costs incurred for the production setup or ordering for each lot of products or raw materials. When the products are produced in lot in a production system, setup costs which include both real out-of- pocket costs and loss of production time are incurred in changing from the production of one product to another. When raw materials are purchased, ordering costs which may include managerial and clerical costs in preparing the purchase order will be incurred.

(c) Shortage costs

These are the costs associated with unsatisfied demand due to a shortage in stock, and take the form of either loss of sales or backorder costs. When sales are lost, both the profit and customer goodwill are lost. If the customers agree to backorder, extra costs are incurred in the loss of customer goodwill, in administrating the backorder, and in penalties for late delivery.

Besides the above-mentioned costs, when the production costs (or purchasing prices) vary with lot size, these costs will also be relevant to and must be considered in the calculation of inventory policies.

There are two decisions to be made in determining every inventory policy : when, and how much a lot is produced (or ordered). An inventory policy is said to be optimal when it satisfies the demand with minimum total inventory costs.

Broadly speaking, there are two major categories of inventory models, namely, static and dynamic models. For the static models, it is assumed that all the parameters are constant. On the other hand, one or more of the parameters in the dynamic model are allowed to be time-varying. Based on different assumptions on the parameters, researchers have developed a vast number of inventory models. Aggarwal [2] has identified the following parameters :

1. Types of demands : constant or time-varying, distribution known or unknown;
2. Costs functions (production costs and inventory costs) : constant, linear, concave or convex;
3. Leadtimes : zero, known or unknown distribution, or dependent on other parameters;
4. Amounts of order quantity actually received : one hundred percentage or less than one hundred percentage with known or unknown distribution;
5. Interdependence between demands of items : single-item or multi-item;
6. Interdependence between locations and/or echelons : single-location,multilocation and/or single-location, multiechelons;
7. Discounting of future costs : constant or variable discounting factors;
8. Various types of constraints : storage, capital, production capacity, etc;

9. Types of backlogging policies : no backlogging, partial backlogging or complete backlogging;
10. Perishability of products : products deteriorate or not deteriorate with time;
11. Types of planning horizon : single-period, multi-period or infinite;
12. Review methods : continuous or periodic review of inventory.

A classical dynamic inventory problem, the dynamic lot size problem, is set up to calculate optimal production (or order) quantities over a given planning horizon when the demand is in discrete lots of time-varying amounts.

1.2 Thesis Overview

This thesis seeks to study and improve the Wagner-Whitin algorithm [25], a long-standing classical approach which solves for optimal solution for the uncapacitated single-item dynamic lot size problem with no backlogging. The Wagner-Whitin algorithm forms a basis for research in this problem. Since its publication in 1958, several modifications and extensions of the algorithm have been developed to solve for the more general cases of the dynamic lot size problem.

In this thesis, a set of network constraints (necessary conditions for optimality) for the Wagner-Whitin algorithm are developed. The validity of these conditions under different cost assumptions are studied. Implementations of these conditions into the algorithm to improve its computational efficiency are considered.

Finally, a simulation experiment is set up. The purposes of this experiment are twofold. First, it examines and compares the performance of the Wagner-Whitin algorithm using different implementations of the network constraints. Second, it studies the performance of the algorithm under different characteristics of the dynamic lot-size problem with constant inventory holding and setup costs, and with no backlogging. Several researchers, for example, Kaimann [18], Berry [6], Ritchie [22], and Gaither [15] have tested the performance of heuristic rules under different problem characteristics. However, little effort has been directed toward the study of the Wagner-Whitin algorithm. Through this experiment, we wish to fill this gap and to provide more insight in the understanding of the Wagner-Whitin algorithm.

CHAPTER II

LITERATURE REVIEW

2.1 Overview of Research Efforts

Optimal lot sizing is one of the fundamental research areas in production management. The dynamic lot size problem is set up to solve for optimal production quantities over a given planning horizon, when the demand is assumed to be in discrete lots of varying amounts.

Dynamic lot size problem has received wide attention in literature. Basically, there are two approaches to solve the problem:

(1) Mathematical models which calculates an optimal solution to the problem as formulated; and

(2) Heuristic rules which generate quick and reasonably good solutions.

The classical and probably earliest lot size formula is the economic order quantity (EOQ) formula, which is also known as the "Wilson formula" [1]. It is not a dynamic lot size model since the demand is assumed to be constant and continuous in this model.

In 1950, Wagner and Whitin [25] published their classical paper "Dynamic Version of the Economic Lot Size Model", which solved for an optimal solution to the dynamic lot size problem. Since their publication, a few modifications and extensions have been added to the original Wagner-Whitin algorithm. In 1968, Zangwill [29, 31] proposed a new approach to the dynamic lot size problem by representing the Wagner-Whitin model as a single source network.

Although the Wagner-Whitin algorithm solves for optimal solutions, it is also considered as "complicated, little understood, and requiring a lot of computer time" [4]. Many researchers and practitioners have thus directed their effort to the development of efficient heuristic rules which calculate solutions close to the optimal ones. As there is an explosive growth in the number of heuristics [22], some researchers became interested in evaluating the performance characteristics of these heuristics (e.g. Kaimann [18], Berry [6] and Ritchie [21]).

2.2 EOQ Model

The earliest known simple lot size formula, the EOQ model, was developed by Ford Harris in 1915. Apparently, it was again independently derived by R.H. Wilson who popularized it. In his honour, it is also referred to as

Wilson formula [1].

In this model, the following assumptions are made:

(1) Demand is deterministic and at a constant rate.

(2) Production costs, inventory holding costs and setup costs are constant.

(3) No backlogging is permitted.

(4) Lead time is zero, ie. production and delivery is instantaneous.

Based on differential calculus, the economic order quantity, Q, is calculated as

$$Q = \sqrt{2sD/h} \tag{2.1}$$

where Q = economic order quantity;

D = demand rate;

s = setup costs;

h = inventory holding costs.

Several modifications to this model have been developed to allow for backlogging , constant lead time and quantity discount [1, 24].

2.3 Wagner-Whitin Algorithm

Wilson Formula for economic lot size will no longer assure minimum cost solution when demand, inventory costs (holding costs, setup costs and shortage costs) and/or production costs vary from period to period over the planning horizon. Under such condition, which is referred as the dynamic version of the model, a more sophisticated mathematical approach is necessary.

Wagner and Whitin [25] developed an algorithm, based on dynamic programming, to solve for the uncapacitated dynamic lot size for single-item product subject to discrete, time-varying demand when the inventory holding and setup costs also vary from period to period. As an order must be placed in the first period (no backordering allowed), there are 2^{n-1} combinations of either ordering or not ordering in each other period for the optimization problem, where n is the number of periods in the planning horizon. However, by formulating the problem into a dynamic program, the computation can be greatly reduced. Their formulation is as follows:

$$F(t) = \min\Big[\min_{1 \le k < t} \{ s_k + \sum_{i=k}^{t-1} \sum_{j=i+1}^{t} h_i D_j + F(k-1) \}, \quad s_t + F(t-1) \Big] \qquad (2.2)$$

where $F(t)$ = minimum cost program for period 1 through t;

s_t = inventory setup cost in period t;

h_t = inventory holding cost in period t;

D_t = demand in period t;

$F(0) = 0$, and $F(1) = s_1$.

The minimum cost program for the problem is given by $F(n)$, where n is the number of periods in the planning horizon. Ordering quantities, Q_t's, can then be calculated based on the minimum cost programs, $F(t)$'s. Under this formulation, only $n(n+1)/2$ possibilities need to be considered.

2.4 Extensions and Modifications to the Wagner-Whitin Algorithm

The original Wagner-Whitin (W-W) model considered only time-varying demand, inventory holding costs and setup costs, and the production cost is assumed to be constant. In 1964, Zabel [29] extended this model to allow for variable production costs. He also developed a backward planning horizon theorem to efficiently solve for the relaxed problem. Later, Zangwill [30, 32] modified the W-W model to permit backlogging of unsatisfied demand.

Eppen, Gould and Pashigian [14] considered the dynamic lot size problem when the inventory holding costs, setup costs and production costs are all time-varying. They developed the General Planning Horizon Theorem that lessens the computational effort to find the optimal solution. Their approach is also based on dynamic programming, and is considered as an extension of the W-W model.

Blackburn and Kunreuther [7] formulated a generalized model for the case when all the costs are time-varying, and developed a procedure to partition the planning horizon for their model which permits backlogging. They showed that both Zangwill and Eppen-Gould-Pashigian's models are special cases of theirs. Their model can be considered as a generalization of W-W model.

2.5 Network Approach

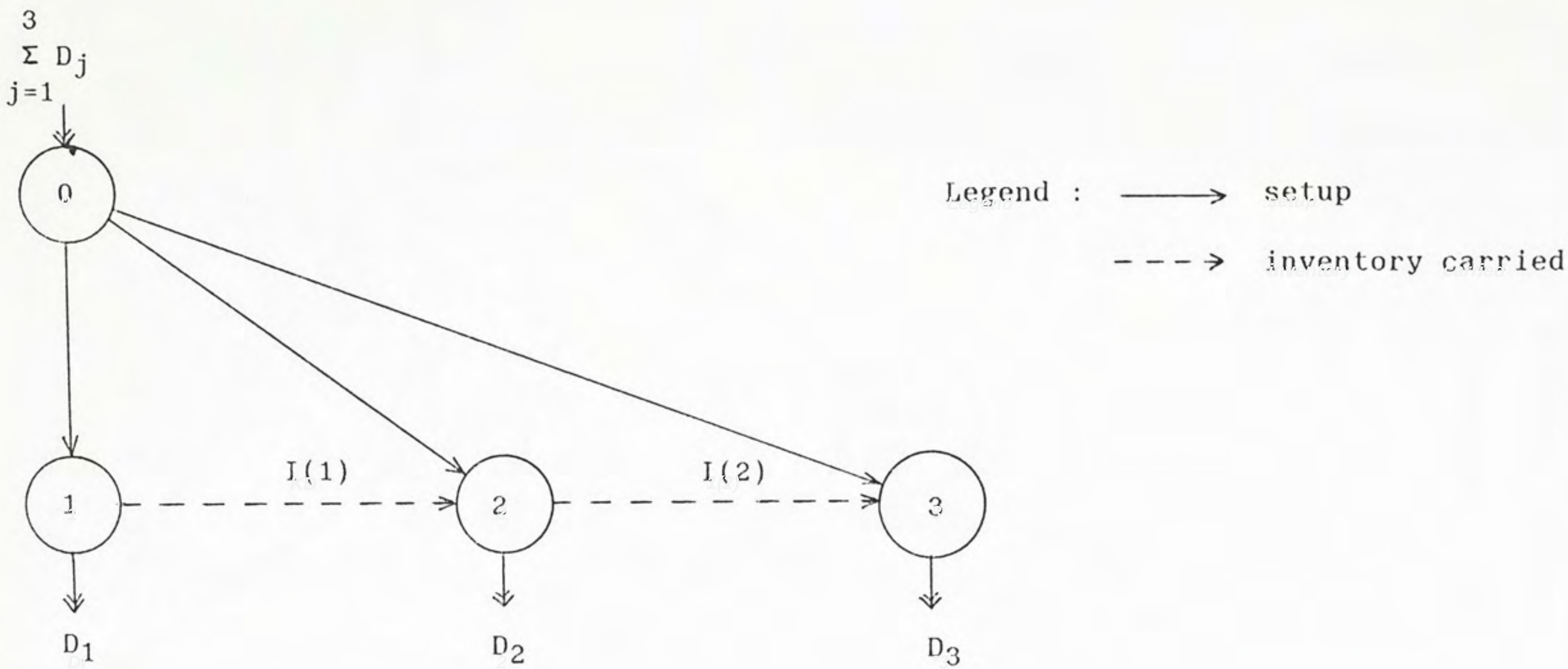
Zangwill [32] represented the Wagner-Whitin model as a single source network. He further improved his network representation to allow for backlogging, and applied the concepts of concave cost network analysis to solve the problem. Zangwill's network representation is depicted in Figure 2.1.

Zangwill's concept of network representation was extended by Chirakiti [12] to show explicitly all the possible inventory levels over the planning horizon. Chirakiti's representation is shown in Figure 2.2.

For a problem with planning horizon of n periods, Chirakiti's network will contain n(n+1)/2 + 1 nodes, Each node represents a possible inventory level, I(i,k), in period i, where I(i,k) stands for the inventory level in period i which will satisfy the demand of the i^{th} period through k^{th} period. Therefore,

$$I(i,k) = \sum_{j=i}^{k} D_j \ , \quad 1 \le i \le n.$$

Obviously, the inventory level in i^{th} period, I(i,k), can take on (n-i+1) possible values. An arc in the network may represent either setup costs or holding costs. In general, an arc joining I(i-1,i-1) to I(i,k) for $i \le k \le n$ will have a setup cost of s_{t-1}, while an arc joining I(i,k) to I(i+1,k) for $i \le k \le n$ will have a holding cost of $h_i I(i+1,k)$.



Note : 1. Each node represents a time period.
2. $I(t)$ represents the amount of inventory carried from period t to period $t+1$.

Figure 2.1 Zangwill's Network Representation of a 3-Period Dynamic Lot Size Problem

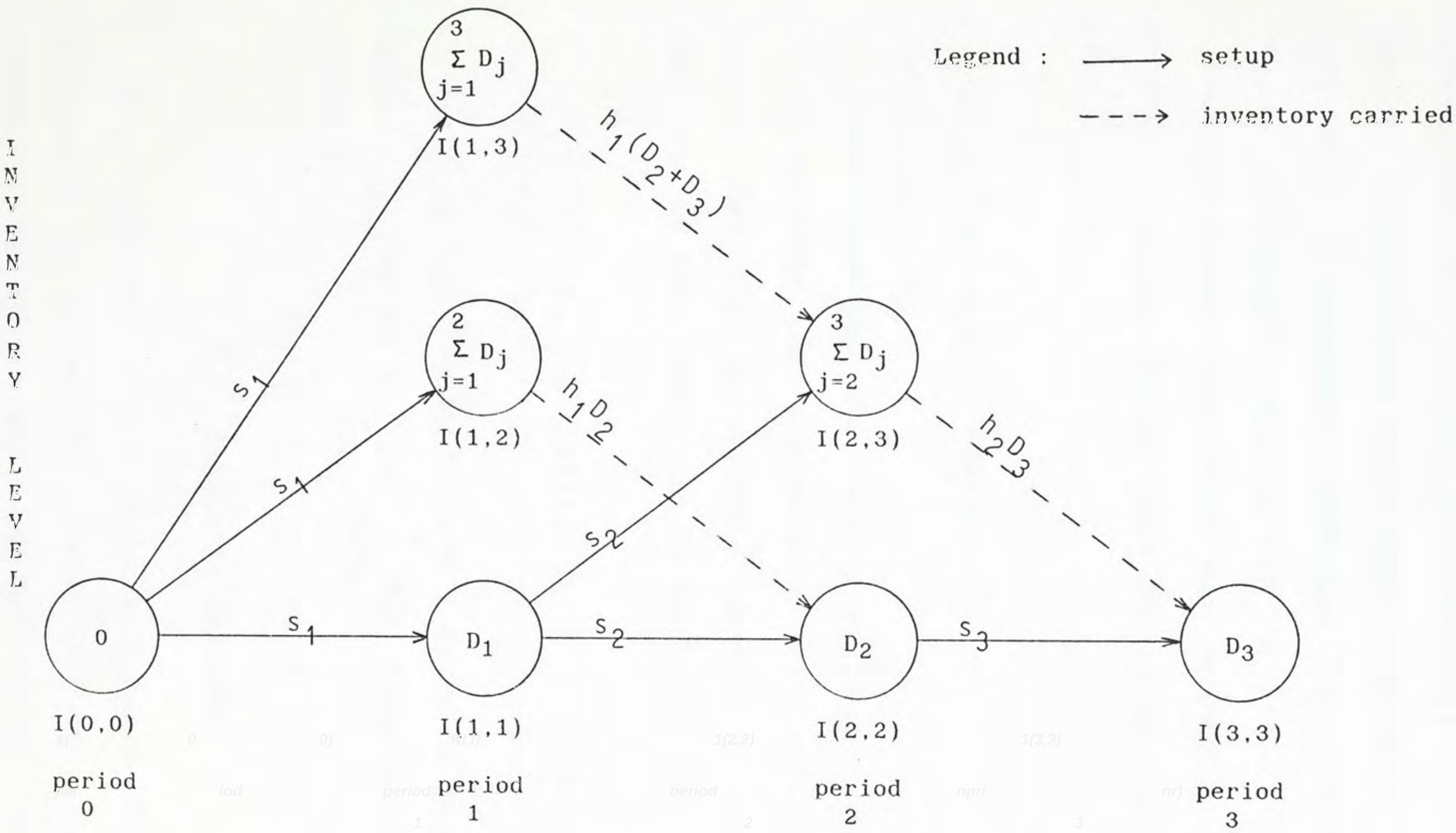


Figure 2.2 Chirakiti's Network Representration of a 3-Period Dynamic Lot Size Problem

Chirakiti's network representation formulated the dynamic lot size problem as a shortest path problem. To improve computational efficiency, Chirakiti developed a constraint (a necessary condition for optimality) which can eliminate part of the set of I(i,k) values. The constraint states that I(i,k) can be part of the optimal solution (ie. one of the nodes on the shortest path) only if

$$h_i \sum_{j=i+1}^{t} D_j \le s_{i+1} \ , \quad 1 \le i < t \le n.$$

Chirakiti proved the validity of this constraint based on the argument that any subpath of an optimal path in a network must also be optimal. However, it can be easily seen that if this constraint is violated, ie.

$$h_i \sum_{j=i+1}^{t} D_j > s_{i+1} \ , \quad 1 \le i < t \le n \quad ,$$

it would be less costly to have a setup in period i+1 for demand D_{i+1} up to D_t (ie. $\sum_{j=i+1}^{t} D_j$) than to hold them in period i .

2.6 Heuristic Methods

Many academic researchers and practitioners have considered the Wagner-Whitin algorithm to be complicated, difficult to understand, and requiring both lengthy computations and large computer storage (eg. Aucamp [4], Gaither [15] and Ritchie [22]). Consequently, simple heuristic rules are developed to calculate approximate solutions to the dynamic lot size problem.

A large number of heuristic rules are found in literature. Among the well-known heuristic rules which can be found easily in most texts [17] are: Economic Order Quantity (EOQ), Period Order Quantity (POQ), Lot-for-Lot, Part Period Balancing (PPB), PPB with look-ahead/look-back and the Silver-Meal algorithm. Some more recent heuristic rules are Incremental Order Quantity Rule [10], Maximum Part-Period Gain Rule [19] and the methods by Gaither [15] and Groff [16].

2.6.1 Classification of Heuristic Methods

Although there are a large number of heuristic rules for dynamic lot size in literature, inevitably, most of them are all based on a few basic principles. It is much easier and convenient to grasp these principles than to understand all the distinct heuristic methods. Ritchie [22] has proposed to classify the existing heuristic rules into three groups. However, a few heuristic rules which are either derived from or closely related to facility location algorithms have not been included in his framework. Naturally, they constitute the fourth groups.

(a) Group 1: Minimum holding and setup costs over the replenishment period

For heuristic rules in this group, the decision to order or not in a period is made based on the principle of minimizing the sum of holding and setup costs over the replenishment interval. These rules may not solve for optimal solutions since only a few demands are considered in making each ordering decision. Two well-known methods belonging this group are those by Silver-Meal [23] and Groff [16].

(b) Group 2: Equal setup and holding costs

Rules in this group calculate lot sizes by equating setup and holding costs in the solutions. This rationale of equating setup and holding costs is consistent with the EOQ model. PPB and De Matties's method [13] are typical examples of this kind of decision rules.

(c) Group 3: Incremental costs

In each period, there are only two mutually exclusive and complementary alternatives - order or not order. The incremental approach calculates the incremental costs associated with each of these two alternatives and selects the one with the lower costs. This approach is adopted by the Incremental Order Quantity method [10] and Gaither' method [15].

(d) Group 4: Fixed-charge facility algorithms

Bahl and Zionts [5], and Law and Khumawala [20] have attempted to solve for dynamic lot size using heuristic rules originally developed for the uncapacitated warehouse location problem. This class of rules can handle variable inventory holding costs and setup costs, as well as backlogging. This is a unique characteristic not found in any other group of heuristic rules. Their performance, in terms of cost deviation from the optimal solutions, compares well with other's. However, due to the flexibility in handling time-varying costs, they are expected to require more computational time in solving problems with constant costs.

2.6.2 Comparison of Heuristic Rules' Performance

In spite of the large number of heuristic rules available, few of them are used in practice [26]. While the reason for such a phenomenon is not clear, it is obvious that choosing an appropriate heuristic rule is difficult. Due to the rapid growth in the number of heuristic rules, their performance characteristics have not been adequately tested and compared.

Most of the available heuristic rules are applicable only to dynamic lot size when the production costs, inventory holding costs and setup costs are constant, and backlogging is not allowed. As a result, comparison of the heuristic rules are also limited under these assumptions.

Several researchers have compared the performance of heuristic rules and reported their results. Kaimann [18] tested the performance of the EOQ method for dynamic lot size based on a set of twenty five problem. Berry [6] compared EOQ, POQ and PPB using Kaimann's data and proposed a more complete analytical framework for comparing the heuristic rules. Most researchers who have then developed new heuristic rules compared their rules with others based on Berry's framework. Ritchie [22] classified the existing heuristic rules into three groups and compared some of the most recent rules.

Heuristic rules are compared and evaluated based on their computational efficiency and the cost deviation of their solutions from the optimal ones. It is found that their performance varies with the characteristics of the problem [6, 15, 18, 22]. Several factors have been identified which characterize a dynamic lot size problem when the various costs are constant and backlogging is not permitted:

(a) Coefficient of Variation (CVD)

The coefficient of variation of the demand is the standard deviation in demand divided by the average demand. It measures the degree of period-to-period variation in demand.

(b) Inventory Costs Ratio (s/h)

This is the ratio of setup costs to inventory holding costs, which affects the frequency of making order. The higher the ratio, the less frequent an order is placed since it will be more costly to make an order.

(c) Average Time Between Orders (TBO)

This is the ratio between EOQ and the average demand. Specifically, TBO is given by

$$\begin{aligned} TBO &= EOQ/\bar{D} \\ &= \sqrt{(2s\bar{D}/h)}/\bar{D} \\ &= \sqrt{2s/(h\bar{D})} \end{aligned}$$

When the demand is uniform , the optimal lot size is given by EOQ and TBO indicates the number of periods covered by each order. When the demand is time-varying, TBO indicates the frequency of making order. When TBO is large, less orders will be placed.

(d) Length of Placing Horizon (n)

To make due comparison among heuristic rules, the length of horizon must be adequate. If it is not so, an inferior rule might produce low cost solution simply because its calculated inventory policy covers exactly to the horizon. Wemmerlor and Whybark [28] suggested a horizon of at least three times TBO for a valid comparison of heuristic rules. The relationship between computational time and n is also different for different rules.

CHAPTER III

NETWORK CONSTRAINTS AND THEIR APPLICATIONS TO THE WAGNER-WHITIN ALGORITHM

3.1 Mathematical Model of the Dynamic Lot Size Problem

The mathematical model for the uncapacitated, single-item dynamic lot size model with no backlogging can be written as follows:

$$\text{Minimize} \quad \sum_{t=1}^{n} \{s_t Y_t + h_t I_t + c_t Q_t\}$$

Subject to :

$$I_{t-1} + Q_t - D_t = I_t$$

$$I_t \geq 0$$

$$Q_t \geq 0$$

$$Y_t = \begin{cases} 1 & \text{if } Q_t > 0, \\ 0 & \text{otherwise.} \end{cases}$$

where s_t = setup costs in period t;

h_t = inventory holding costs in period t;

c_t = production costs in period t;

I_t = inventory level in period t;

Q_t = order/ production quantity in period t;

Y_t = indicator variable which takes on the value of one when there is an order (or production) in period t, and takes on the value of zero when otherwise.

3.2 Generalized Network Representation

In Chirakiti's network representation, only the inventory setup costs and holding costs are variable, while the production costs are held constant. A more generalized network representation of the dynamic lot size problem when production costs are also time-varying is depicted in Figure 3.1.

As before, the inventory level in the i^{th} period can take on (n-i+1) possible values:

$$I(i,k) = \sum_{j=i}^{k} D_j , \quad 1 \le i \le k \le n.$$

I(i,k) represents the inventory level containing the demands of the i^{th} period through the k^{th} period. Here, an arc joining I(i-1,i-1) to I(i,k) for $i \le k \le n$ will incur a cost of $s_i + c_i I(i,k)$, while an arc joining I(i,k) to I(i+1,k) will incur a cost of $h_i I(i+1,k)$.

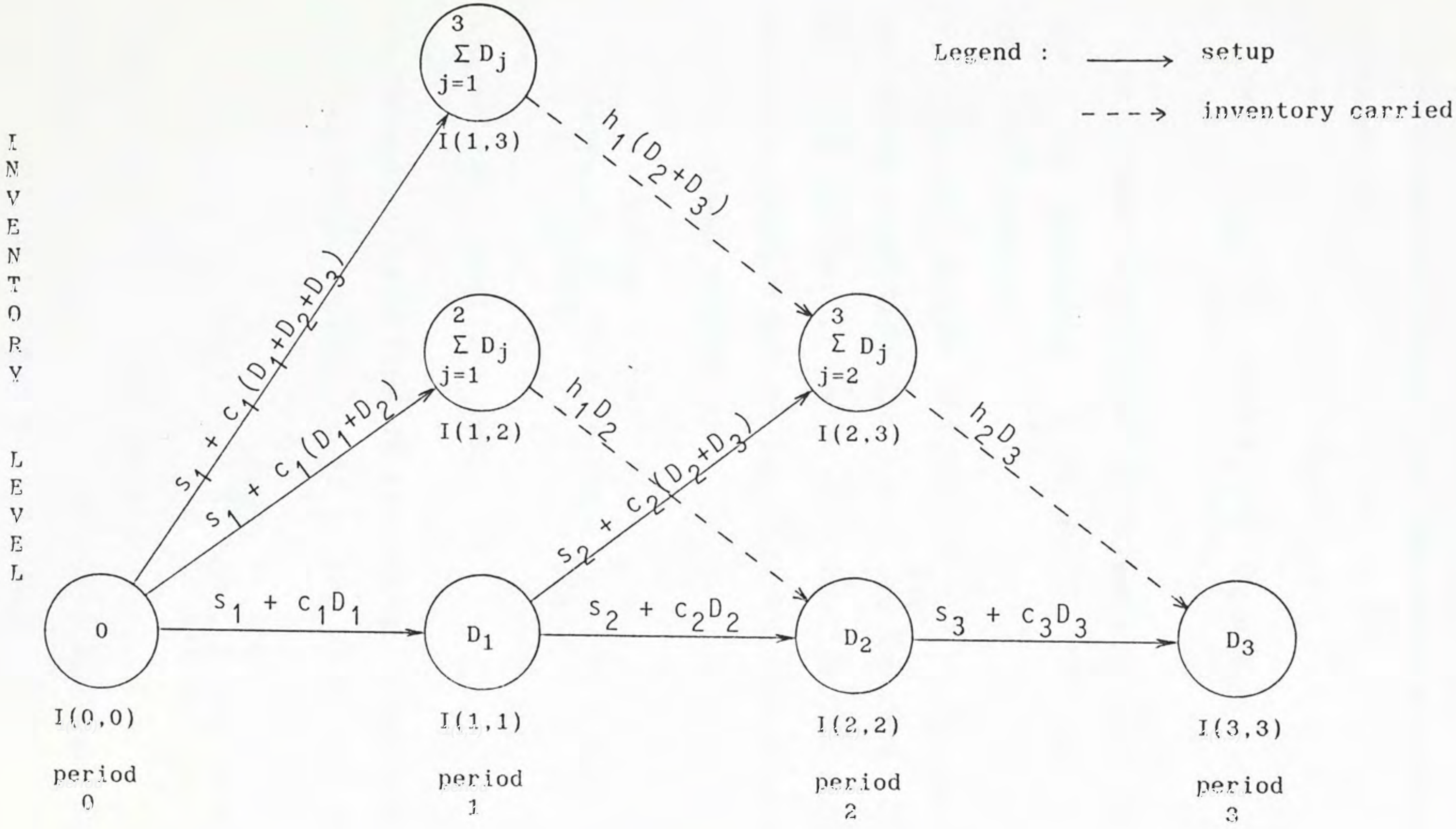


Figure 3.1 Generalized Network Representration of a 3-Period Dynamic Lot Size Problem

3.3 Generalized Network Constraints

Referring to Figure 3.1, Chirakiti's constraint can be restated as

$$(c_i - c_{i+1}) \sum_{j=i+1}^{t} D_j + h_i \sum_{j=i+1}^{t} D_j \le s_{i+1} \quad , \; 1 \le i < t \le n. \quad (3.1)$$

The first term is the difference between the production costs for producing $\sum_{j=i+1}^{t} D_j$ in the i^{th} period and that in the $i+1^{th}$ period.

Consider again $I(i,t) = \sum_{j=i}^{t} D_j$. If $t \ge i+2$, then a necessary condition for $I(i,t)$ to be part of the optimal solution is that it is no more costlier to hold $\sum_{j=i+2}^{t} D_j$ in period i through period i+1 than to have a setup in period i+2 for these demand, or

$$(c_i - c_{i+2}) \sum_{j=i+2}^{t} D_j + (h_i + h_{i+1}) \sum_{j=i+2}^{t} D_j \le s_{i+2},$$

where $i \ge 1$, and $i+2 \le t \le n$. (3.2)

Extending this line of reasoning, a general form for the constraints given in (3.1) and (3.2) can written as

$$(c_i - c_{i+m}) \left(\sum_{j=i+m}^{t} D_j \right) + \left(\sum_{g=i}^{i+m-1} h_g \right) \left(\sum_{j=i+m}^{t} D_j \right) \le s_{i+m},$$

where $m \ge 1$, $i \ge 1$, and $m+i \le t \le n$. (3.3)

A formal proof of (3.3) is given in the next section .

3.4 Relationship Between Network Constraints and the Wagner-Whitin Algorithm

When the production costs, inventory holding costs and setup costs are all time-varying, the Wagner-Whitin forward algorithm can be written as:

$$F(t) = \min \{ \min_{0 \le k < t-1} [F(k) + s_{k+1} + \sum_{j=k+2}^{t} (\sum_{i=k+1}^{j-1} h_i) D_j + c_{k+1} \sum_{j=k+1}^{t} D_j], F(t-1) + s_t + c_t D_t \}$$

$$= \min_{0 \le k < t-1} [F(k) + s_{k+1} + \sum_{j=k+2}^{t} (\sum_{i=k+1}^{j-1} h_i) D_j + c_{k+1} \sum_{j=k+1}^{t} D_j],$$

since the expression in the bracket reduces to

$F(t-1) + s_t + c_t D_t$ when $k=t-1$.

Let $A(k,t) = \sum_{j=k+2}^{t} (\sum_{i=k+1}^{j-1} h_i) D_j$

= inventory holding costs incurred when the quantity $\sum_{j=k+2}^{t} D_j$ is produced in period $k+1$;

$I(k+1,t) = \sum_{j=k+1}^{t} D_j$

= inventory level in period $k+1$ which covers demand from period $k+1$ through period t;

$$T(k,t) = F(k) + s_{k+1} + A(k,t) + c_{k+1}I(k+1,t)$$

= total costs incur for the k^{th} inventory policy to cover period 1 through period t, where $0 \le k \le t-1$;

then the forward algorithm can be written as :

$$F(t) = \min_{0 \le k \le t-1} T(k,t).$$

Suppose $F(t) = T(k_0,t)$, then $I(k_0+1,t)$ must satisfy (3.3). By replacing i in (3.3) with k_0+1, k_0 must satisfy the following $(t-k_0-1)$ inequalities:

$$\left(\sum_{i=k_0+1}^{k_0+m} h_i + c_{k_0+1} - c_{k_0+m+1}\right)\left(\sum_{j=k_0+m+1}^{t} D_j\right) \le s_{k_0+m+1},$$

where $m \ge 1$, $k_0 \ge 0$, and $m+k_0+1 \le t \le n$. (3.4)

A proof of (3.4) is given below:

<u>Proof</u>

Given $F(t) = T(k_0,t)$, $\therefore$ $T(k_0,t) \le T(k,t)$, $0 \le k \le t-1$.

For $1 \le m \le t-k_0-1$,

$$F(k_0+m) = \min_{0 \le l \le k_0+m-1} T(l,k_0+m)$$

$$\le T(k_0,\ k_0+m)\ , \quad \text{as} \quad 0 \le k_0 \le k_0+m-1. \qquad (3.5)$$

Suppose $\exists\ m_0$ be s.t.

(a) $1 \le m_0 \le t-k_0-1$

(b) $\left(\sum_{i=k_0+1}^{k_0+m_0} h_i + c_{k+1} - c_{k+m+1}\right)\left(\sum_{j=k_0+m_0+1}^{t} D_j\right) > s_{k+m+1}$

(3.6)

From (b) in (3.6) above,

$$\left(\sum_{i=k_0+1}^{k_0+m_0} h_i + c_{k_0+1} - c_{k_0+m_0+1}\right)\left(\sum_{j=k_0+m_0+1}^{t} D_j\right)$$

$$> s_{k_0+m_0+1}$$

$$\Rightarrow \left(\sum_{i=k_0+1}^{k_0+m_0} h_i\right)\left(\sum_{j=k_0+m_0+1}^{t} D_j\right) + c_{k_0+1}\left(\sum_{j=k_0+m_0+1}^{t} D_j\right)$$

$$> s_{k_0+m_0+1} + c_{k_0+m_0+1}\left(\sum_{j=k_0+m_0+1}^{t} D_j\right)$$

$$\Rightarrow \sum_{j=k_0+m_0+1}^{t}\left(\sum_{i=k_0+1}^{k_0+m_0} h_i\right)D_j + c_{k_0+1}\left(\sum_{j=k_0+m_0+1}^{t} D_j\right)$$

$$> s_{k_0+m_0+1} + c_{k_0+m_0+1}\left(\sum_{j=k_0+m_0+1}^{t} D_j\right). \qquad (3.7)$$

Now consider the following :

$$T(k_0,t)$$

$$= F(k_0) + s_{k_0+1} + A(k_0,t) + c_{k_0+1}I(k_0+1,t)$$

$$= F(k_0) + s_{k_0+1} + \sum_{j=k_0+2}^{t}\left(\sum_{i=k_0+1}^{j-1} h_i\right)D_j + c_{k_0+1}\left(\sum_{j=k_0+1}^{t} D_j\right)$$

$$= F(k_0) + s_{k_0+1} + \sum_{j=k_0+2}^{k_0+m_0}\left(\sum_{i=k_0+1}^{j-1} h_i\right)D_j$$

$$+ \sum_{j=k_0+m_0+1}^{t}\left(\sum_{i=k_0+1}^{j-1} h_i\right)D_j + c_{k_0+1}\left(\sum_{j=k_0+1}^{k_0+m_0} D_j\right)$$

$$+ c_{k_0+1}\left(\sum_{j=k_0+m_0+1}^{t} D_j\right)$$

$$= F(k_0) + s_{k_0+1} + A(k_0,k_0+m_0) + c_{k_0+1}I(k_0+1,k_0+m_0)$$

$$+ \sum_{j=k_0+m_0+1}^{t} \left(\sum_{i=k_0+1}^{j-1} h_i \right) D_j + c_{k_0+1} \left(\sum_{j=k_0+m_0+1}^{t} D_j \right)$$

$$= T(k_0, k_0+m_0) + \sum_{j=k_0+m_0+1}^{t} \left(\sum_{i=k_0+1}^{j-1} h_i \right) D_j + c_{k_0+1} \left(\sum_{j=k_0+m_0+1}^{t} D_j \right)$$

$$\geq F(k_0+m_0) + \sum_{j=k_0+m_0+1}^{t} \left(\sum_{i=k_0+1}^{j-1} h_i \right) D_j + c_{k_0+1} \left(\sum_{j=k_0+m_0+1}^{t} D_j \right)$$

[by (3.5)]

$$= F(k_0+m_0) + \sum_{j=k_0+m_0+1}^{t} \left(\sum_{i=k_0+1}^{k_0+m_0} h_i \right) D_j$$

$$+ \sum_{j=k_0+m_0+1}^{t} \left(\sum_{j=k_0+m_0+1}^{j-1} h_i \right) D_j + c_{k_0+1} \left(\sum_{j=k_0+m_0+1}^{t} D_j \right)$$

$$= F(k_0+m_0) + \sum_{j=k_0+m_0+1}^{t} \left(\sum_{i=k_0+1}^{k_0+m_0} h_i \right) D_j$$

$$+ \sum_{j=k_0+m_0+2}^{t} \left(\sum_{j=k_0+m_0+1}^{j-1} h_i \right) D_j + c_{k_0+1} \left(\sum_{j=k_0+m_0+1}^{t} D_j \right)$$

[by (*) below]

$$= F(k_0+m_0) + \sum_{j=k_0+m_0+2}^{t} \left(\sum_{i=k_0+m_0+1}^{j-1} h_i \right) D_j$$

$$+ \sum_{j=k_0+m_0+1}^{t} \left(\sum_{j=k_0+1}^{k_0+m_0} h_i \right) D_j + c_{k_0+1} \left(\sum_{j=k_0+m_0+1}^{t} D_j \right)$$

$$= F(k_0+m_0) + A(k_0+m_0, t) + \sum_{j=k_0+m_0+1}^{t} \left(\sum_{i=k_0+1}^{k_0+m_0} h_i \right) D_j$$

$$+ c_{k_0+1}\left(\sum_{j=k_0+m_0+1}^{t} D_j\right)$$

$$> F(k_0+m_0) + A(k_0+m_0,t) + s_{k_0+m_0+1} + c_{k_0+m_0+1}\left(\sum_{j=k_0+m_0+1}^{t} D_j\right)$$

[by (3.7)]

$$= T(k_0+m_0,t)\ ,\quad 0<k_0+1\le k_0+m_0\le t-1$$

A contradiction since $T(k_0,t) \le T(k,t)$ for $0\le k\le t-1$. Therefore, (3.6) must not be correct.

ie. $\forall m$, where $1\le m\le t-k_0-1$,

$$\left(\sum_{i=k_0+1}^{k_0+m} h_i + c_{k_0+1} - c_{k_0+m+1}\right)\left(\sum_{j=k_0+m+1}^{t} D_j\right) \le s_{k_0+m+1}.$$

Note: (*)

$$\sum_{j=k_0+m_0+1}^{t}\left(\sum_{i=k_0+m_0+1}^{j-1} h_i\right)D_j = \left(\sum_{i=k_0+m_0+1}^{k_0+m_0+1-1} h_i\right)D_j$$

$$+ \sum_{j=k_0+m_0+2}^{t}\left(\sum_{i=k_0+m_0+1}^{j-1} h_i\right)D_j$$

$$= 0 + \sum_{j=k_0+m_0+2}^{t}\left(\sum_{i=k_0+m_0+1}^{j-1} h_i\right)D_j$$

$$= \sum_{j=k_0+m_0+2}^{t}\left(\sum_{i=k_0+m_0+1}^{j-1} h_i\right)D_j$$

□

Three particular cases for (3.4) are considered. First, when the production costs are constant, (3.4) can be simplified as:

$$\left(\sum_{i=k+1}^{k+m} h_i\right)\left(\sum_{j=k+m+1}^{t} D_j\right) \le s_{k+m+1}, \quad \text{where } m\ge 1,\ k_0\ge 0,$$

$$\text{and } m+k_0+1\le t\le n. \quad (3.4a)$$

When both the production costs and the inventory costs are constant, (3.4) can be further simplified as:

$$mh\left(\sum_{j=k+m+1}^{t} D_j\right) \le s_{k+m+1}, \quad \text{where } m\ge 1,\ k_0\ge 0,$$

$$\text{and } m+k_0+1\le t\le n. \qquad (3.4b)$$

When all the three costs are constant, (3.4) can be written as:

$$mh\left(\sum_{j=k+m+1}^{t} D_j\right) \le s, \quad \text{where } m\ge 1,\ k_0\ge 0,$$

$$\text{and } m+k_0+1\le t\le n. \qquad (3.4c)$$

3.5 Lower Bounds for the Wagner-Whitin Algorithm

The original Wagner-Whitin algorithm is written as

$$F(t) = \min_{0\le k\le t-1} T(k,t) \qquad (3.8)$$

If $F(t) = F(k_0)$, then k_0 must satisfy a set of inequalities as given in (3.4). Based on these inequalities, it is possible to calculate lower bounds for k in (3.8).

A lower bound, l_t, for k exists when k does not satisfy the optimality conditions given in (3.4) for all $k < l_t$. However, this would imply that $F(t) = F(k)$,

$0 \le k \le l_t$. Therefore, if l_t exists, (3.8) can be written as

$$F(t) = \min_{l_t \le k \le t} T(k,t) \tag{3.8a}$$

<u>Theorem</u>

If $c_{i+1} - c_i \le h_i$, where $1 \le i \le n-1$, then lower bound, l_t, exists.

<u>Proof</u>

Given $c_{i+1} - c_i \le h_i$, $\therefore$ $h_i + c_i - c_{i+1} \ge 0$.

Suppose there exists l $(0 \le l \le t-1)$ which does not satisfy (3.4).

$\therefore \exists m, \quad 1 \le m \le t-l-1$

$$\left(\sum_{i=l+1}^{l+m} h_i + c_{l+1} - c_{l+m+1} \right) \left(\sum_{j=l+m+1}^{t} D_j \right) > s_{l+m+1} \tag{3.9}$$

Now, consider the case for $l-1$. Let $m' = m+1$.

$$m' = m+1$$

$$\therefore \quad 2 \le m' \le t-l-1+1$$

$$= t-(l-1)-1$$

Next, consider

$$\left(\sum_{i=(l-1)+1}^{(l-1)+m'} h_i + c_{(l-1)+1} - c_{(l-1)+m'+1} \right) \left(\sum_{j=(l-1)+m'+1}^{t} D_j \right)$$

$$= \left(\sum_{i=l}^{l+m} h_i + c_l - c_{l+m+1} \right) \left(\sum_{j=l+m+1}^{t} D_j \right)$$

$$= \left(\sum_{i=l+1}^{l+m} h_i + c_{l+1} - c_{l+m+1}\right)\left(\sum_{i=l+m'+1}^{t} D_j\right)$$

$$+ (h_l + c_l + c_{l+1})\left(\sum_{i=l+m+1}^{t} D_j\right)$$

$$> s_{l+m+1} + (h_l + c_l - c_{l+1})\left(\sum_{i=l+m+1}^{t} D_j\right) \qquad \text{[by (3.9)]}$$

$$\geq s_{l+m+1} + 0 \qquad \text{[since } h_i + c_i - c_{i+1} \geq 0 \text{]}$$

$$= s_{(l-1)+m'+1}$$

Therefore, if l does not satisfy (3.4), (l-1) will not satisfy (3.4). By similar argument, (l-2) through 0 will not satisfy (3.4). Hence, l is a lower bound for k. □

<u>Corollary</u>

If production costs are constant, then lower bound for k in (3.8) always exists.

The proof to this corollary is obvious since when production costs are constant,

$$c_i - c_{i+1} = 0, \quad i = 1, \ldots, n-1.$$

$$\therefore c_i - c_{i+1} \leq h_i$$

$$\Leftrightarrow \qquad h_i \geq 0$$

which must be true. □

The lower bound l_t can be found by substituting the sequence {t-1, t-2, ..., 0}, one at a time, into (3.4) until (3.4) is violated. l_t is then given by the smallest number in the sequence which satisfies (3.4).

When the production costs and inventory holding costs are constant, the lower bound, l_t, can be written explicitly in terms of s, h, and D. From (3.4b),

$$mh\left(\sum_{j=k_0+m+1}^{t} D_j\right) \le s_{k+m+1}, \quad \text{where } m \ge 1,\ k_0 \ge 0,$$

$$\text{and } m+k_0+1 \le t \le n.$$

When $m=t-k_0-1$, $(t-k_0-1)hD_t \le s_t$

$$t-k_0-1 \le s_t/(hD_t)$$

$$k_0 \ge t-1-s_t/(hD_t)$$

$$l_t = \max\{0, \mathrm{Int}[t-1-s_t/(hD_t)+1]\}$$

$$= \max\{0, \mathrm{Int}[t-s_t/(hD_t)]\}.$$

When $m=t-k_0-2$, $(t-k_0-2)h(D_{t-1}+D_t) \le s_{t-1}$

$$t-k_0-2 \le s_{t-1}/[h(D_{t-1}+D_t)]$$

$$k_0 \ge t-2-s_{t-1}/[h(D_{t-1}+D_t)]$$

$$l_t = \max\{0, \mathrm{Int}[t-1-s_{t-1}/h(D_{t-1}+D_t]\}.$$

In general, (3.4b) provides a set of $(t-l_t-1)$ values for l_t. Therefore,

$$l_t = \max\{0, \mathrm{Int}[(t-p) - s_{t-p}/(h\sum_{j=t-p}^{t} D_j)]\}, \quad 0 \le p \le t-l_t-2. \tag{3.10}$$

It should be noted that in (3.4), the number of inequalities depends on k_0. Therefore, the number of values for l_t is recursive with l_t. However, when the lower bound calculation is implemented into the Wagner-

Whitin algorithm, this problem can be overcome by a suitable searching procedure for l_t. Starting from t and searching backwards, the procedure will stop when the current l_t is reached in the sequence {t, t-1, ..., 0}. A searching subroutine in BASIC code is listed below:

```
LT = 0
SUM = 0
FOR  I = T  TO  LT
     SUM = SUM + D(I)
     LL = INT(I - S(I)/(H*SUM))
     IF LL > LT  THEN LT = LL
NEXT I
```

3.6 Lower Bound Theorem

An interesting property of the lower bounds, as stated in the theorem below, allows for a very efficient implementation of the network constraints into the Wagner-Whitin algorithm. This property, however, holds only when the production costs are constant.

Theorem

If l_t is a lower bound for k in F(t), then l_t is also a lower bound for k in F(t+1).

Proof

If l_t is a lower bound for k in F(t), then F(t) can written as:

$$F(t) = \min_{l_t \leq k \leq t-1} T(k,t)$$

where $\quad T(k,t) = F(k) + s_{k+1} + A(k,t)$

$$A(k,t) = \sum_{j=k+2}^{t} \left(\sum_{i=k+1}^{j-1} h_i \right) D_j$$

Let $F(t) = T(k_0,t)$. Then $k_0 \geq l_t$, as l_t is a lower bound. Now consider $F(t+1)$ and suppose that l_t is not a lower bound for k in $F(t+1)$. Then $F(t+1) = T(k_1,t+1)$, $0 \leq k_1 < l_t \leq k_0$.

Now,

$$T(k_1,t+1) = F(k_1) + s_{k+1} + A(k_1,t+1)$$

$$= F(k_1) + s_{k+1} + \sum_{j=k_1+2}^{t+1} \left(\sum_{i=k_1+1}^{j-1} h_i \right) D_j$$

$$= F(k_1) + s_{k+1} + \sum_{j=k_1+2}^{t} \left(\sum_{i=k_1+1}^{j-1} h_i \right) D_j + \left(\sum_{i=k_1+1}^{t} h_i \right) D_{t+1}$$

$$= T(k_1,t) + \left(\sum_{i=k_1+1}^{t} h_i \right) D_{t+1}$$

$$\geq T(k_0,t) + \left(\sum_{i=k_1+1}^{t} h_i \right) D_{t+1} \quad [\text{since } F(t)=T(k_0,t)]$$

$$> T(k_0,t) + \left(\sum_{i=k_0+1}^{t} h_i \right) D_{t+1} \quad [\text{since } k_0 \geq l_t > k_1]$$

$$= F(k_0) + s_{k+1} + \sum_{j=k_0+2}^{t} (\sum_{i=k_0+1}^{j-1} h_i)D_j$$

$$+ (\sum_{i=k_0+1}^{t} h_i)D_{t+1}$$

$$= F(k_0) + s_{k+1} + \sum_{j=k_0+2}^{t+1} (\sum_{i=k_0+1}^{j-1} h_i)D_j$$

$$= T(k_0,t+1).$$

This is a contradiction since $F(t+1) = T(k_1,t+1)$

$= T(k_1,t+1) \leq t(k,t+1),\ 0 \leq k \leq t.$

Therefore, the assumption is wrong, ie. l_t is a lower bound for k in F(t+1), or

$$F(t+1) = T(k_1,t+1) \ ,\quad k_1 \geq l_t. \qquad \square$$

The lower bound for F(t), l^*_t, can either be calculated from (3.10) or simply be taken as l^*_{t-1}, based on the lower bound theorem. Therefore, l^*_t can be written as:

$$l^*_t = \max\ (l^*_{t-1},\ l_t),\quad 1 \leq t \leq n;$$

where $l_t = \max\{Int[(t-p) - s_{t-p}/(h \sum_{j=t-p}^{t} D_j)]\},\ 0 \leq p \leq t-l_t-2,$

and $\quad l^*_0 = 0.$

In essence, the lower bound theorem partitions the problem into smaller horizon as the algorithm is proceeded forward, and thereby reduces the computational effort.

3.7 Implementation of Network Constraints

Although (3.10), when fully employed, searches for the largest lower bound for l_t, it also requires extra searching effort. There is a tradeoff between the saving in computational time using a larger lower bound and the extra time required in searching for such a lower bound. The optimal implementation of (3.10) should result in a minimum sum of these two. Here, the following four implementations of (3.10) are considered:

$$l_t = \max\{\mathrm{Int}[(t-p) - s_{t-p}/(h \sum_{j=t-p}^{t} D_j)]\}, \quad 0 \le p \le a$$

Implementation	Value of a
WW1	0
WW2	1
WW3	2
WWT1	$t-l_t-2$

Only WWT1 is the full implementation of (3.10). To illustrate the effectiveness of the network constraints in reducing computations of the Wagner-Whitin algorithm, a demand data set taken from Berry's study [6], which is shown below, is used as an example. The dynamic programming tables for different cases are calculated and shown in tables 3.1 to 3.5.

Demand = {80, 100, 125, 100, 50, 50, 100, 125, 125, 100, 50, 100}

s = 500, h = 1

TABLE 3.1

FULL DYNAMIC PROGRAMMING TABLE

t	1	2	3	4	5	6	7	8	9	10	11	12
D	80	100	125	100	50	50	100	125	125	100	50	100
	500	600	850	1150	1350	1600	2200	3075	4075	4975	5475	6575
		1000	1125	1325	1475	1675	2175	2925	3800	4600	5050	6050
			1170	1200	1300	1450	1850	2475	3225	3925	4325	5225
				1350	1400	1500	1800	2300	2925	3525	3875	4675
					1650	1700	1900	2275	2775	3275	3575	4275
						1800	1900	2150	2525	2925	3175	3775
							1950	2075	2325	2625	2825	3325
								2300	2425	2625	2775	3175
									2575	2675	2775	3075
										2825	2875	3075
											3125	3225
												3275

Total number of entries = 78

TABLE 3.2

DYNAMIC PROGRAMMING TABLE FOR WW1

t	1	2	3	4	5	6	7	8	9	10	11	12
D	80	100	125	100	50	50	100	125	125	100	50	100
l_t	0	0	0	0	0	0	2	4	5	5	5	7
	500	600	850	1150	1350	1600	<u>2200</u>	<u>3075</u>	<u>4075</u>	<u>4975</u>	<u>5475</u>	<u>6575</u>
		1000	1125	1325	1475	1675	<u>2175</u>	<u>2925</u>	<u>3800</u>	<u>4600</u>	<u>5050</u>	<u>6050</u>
			1170	1200	1300	1450	1850	<u>2475</u>	<u>3225</u>	<u>3925</u>	<u>4325</u>	<u>5225</u>
				1350	1400	1500	1800	<u>2300</u>	<u>2925</u>	<u>3525</u>	<u>3875</u>	<u>4675</u>
					1650	1700	1900	2275	<u>2775</u>	<u>3275</u>	<u>3575</u>	<u>4275</u>
						1800	1900	2150	2525	2925	3175	<u>3775</u>
							1950	2075	2325	2625	2825	<u>3325</u>
								2300	2425	2625	2775	3175
									2575	2675	2775	3075
										2825	2875	3075
											3125	3225
												3275

Note: 1. $l_t = \max\{l_{t-1}, l_t\}$, where $l_0=0$ and

$$l_t = \max\{\mathrm{Int}[(t-p) - s_{t-p}/(h \sum_{j=t-p}^{t} D_j)]\}, \; p=0.$$

2. Underlined entries are eliminated via the lower bound calculation and the lower bound theorem.

Total number of entries = 50

TABLE 3.3

DYNAMIC PROGRAMMING TABLE FOR WW2

t	1	2	3	4	5	6	7	8	9	10	11	12
D	80	100	125	100	50	50	100	125	125	100	50	100
l_t	0	0	0	0	0	0	2	4	6	6	6	7
	500	600	850	1150	1350	1600	$\underline{2200}$	$\underline{3075}$	$\underline{4075}$	$\underline{4975}$	$\underline{5475}$	$\underline{6575}$
		1000	1125	1325	1475	1675	$\underline{2175}$	$\underline{2925}$	$\underline{3800}$	$\underline{4600}$	$\underline{5050}$	$\underline{6050}$
			1170	1200	1300	1450	1850	$\underline{2475}$	$\underline{3225}$	$\underline{3925}$	$\underline{4325}$	$\underline{5225}$
				1350	1400	1500	1800	$\underline{2300}$	$\underline{2925}$	$\underline{3525}$	$\underline{3875}$	$\underline{4675}$
					1650	1700	1900	2275	$\underline{2775}$	$\underline{3275}$	$\underline{3575}$	$\underline{4275}$
						1800	1900	2150	$\underline{2525}$	$\underline{2925}$	$\underline{3175}$	$\underline{3775}$
							1950	2075	2325	2625	2825	$\underline{3325}$
								2300	2425	2625	2775	3175
									2575	2675	2775	3075
										2825	2875	3075
											3125	3225
												3275

Note: 1. $l_t = \max\{l_{t-1}, l_t\}$, where $l_0=0$ and

$$l_t = \max\{\mathrm{Int}[(t-p) - s_{t-p}/(h \sum_{j=t-p}^{t} D_j)]\},\ 0\le p\le 1.$$

2. Underlined entries are eliminated via the lower bound calculation and the lower bound theorem.

Total number of entries = 47

TABLE 3.4

DYNAMIC PROGRAMMING TABLE FOR WW3

t	1	2	3	4	5	6	7	8	9	10	11	12
D	80	100	125	100	50	50	100	125	125	100	50	100
l_t	0	0	0	0	1	1	2	4	6	6	7	8
	500	600	850	1150	$\underline{1350}$	$\underline{1600}$	$\underline{2200}$	$\underline{3075}$	$\underline{4075}$	$\underline{4975}$	$\underline{5475}$	$\underline{6575}$
		1000	1125	1325	1475	1675	$\underline{2175}$	$\underline{2925}$	$\underline{3800}$	$\underline{4600}$	$\underline{5050}$	$\underline{6050}$
			1170	1200	1300	1450	1850	$\underline{2475}$	$\underline{3225}$	$\underline{3925}$	$\underline{4325}$	$\underline{5225}$
				1350	1400	1500	1800	$\underline{2300}$	$\underline{2925}$	$\underline{3525}$	$\underline{3875}$	$\underline{4675}$
					1650	1700	1900	2275	$\underline{2775}$	$\underline{3275}$	$\underline{3575}$	$\underline{4275}$
						1800	1900	2150	$\underline{2525}$	$\underline{2925}$	$\underline{3175}$	$\underline{3775}$
							1950	2075	2325	2625	$\underline{2825}$	$\underline{3325}$
								2300	2425	2625	2775	$\underline{3175}$
									2575	2675	2775	3075
										2825	2875	3075
											3125	3225
												3275

Note: 1. $l_t = \max\{l_{t-1}, l_t\}$, where $l_0=0$ and

$$l_t = \max\{\mathrm{Int}[(t-p) - s_{t-p}/(h \sum_{j=t-p}^{t} D_j)]\},\ 0 \le p \le 2.$$

2. Underlined entries are eliminated via the lower bound calculation and the lower bound theorem.

Total number of entries = 43

TABLE 3.5

DYNAMIC PROGRAMMING TABLE FOR WWT1

t	1	2	3	4	5	6	7	8	9	10	11	12
D	80	100	125	100	50	50	100	125	125	100	50	100
l_t	0	0	0	0	1	1	2	4	6	6	7	8
	500	600	850	1150	1350	1600	2200	3075	4075	4975	5475	6575
		1000	1125	1325	1475	1675	2175	2925	3800	4600	5050	6050
			1170	1200	1300	1450	1850	2475	3225	3925	4325	5225
				1350	1400	1500	1800	2300	2925	3525	3875	4675
					1650	1700	1900	2275	2775	3275	3575	4275
						1800	1900	2150	2525	2925	3175	3775
							1950	2075	2325	2625	2825	3325
								2300	2425	2625	2775	3175
									2575	2675	2775	3075
										2825	2875	3075
											3125	3225
												3275

Note: 1. $l_t = \max\{l_{t-1}, l_t\}$, where $l_0=0$ and

$$l_t=\max\{\mathrm{Int}[(t-p)-s_{t-p}/(h \sum_{j=t-p}^{t} D_j)]\}, 0\le p\le t-l_t-2.$$

2. Underlined entries are eliminated via the lower bound calculation and the lower bound theorem.

Total number of entries = 43

CHAPTER IV

EXPERIMENTAL STUDIES

4.1 Introduction

Several experimental studies have been carried out on heuristic rules for dynamic lot size problem with constant costs [6, 8, 9, 15, 18, 22]. The purpose is to either rank the performance of different heuristic methods based on computational times and deviations from optimal cost solutions, or test the impact of different factors on the heuristics' performance. Little research is conducted on the performance of the Wagner-Whitin algorithm, in terms of computational times, under different experimental factors. Gaither's study [15], however, indicates a highly exponential relationship between the computational time and the length of planning horizon (n).

In this thesis, a simulation experiment is set up to test the performance of the Wagner-Whitin algorithm when different implementations of the network constraints are incorporated. Production costs, setup costs and inventory holding costs are all held constant. The four implementations discussed in section 3.7 are tested, and the original Wagner-Whitin algorithm is also included as a basis for comparison. As all the implementations solve for optimal solution, computational time is the only measure of performance. Factors that are expected to affect computational time will be analysed. It is also hoped that, through this experiment, some insight into the relationships between the performance of the algorithm and different experimental factors can be provided.

4.2 Experimental Factors

In studying the performance of heuristic rules, the following factors are generally included by most researchers [6, 9, 15, 18, 22] :

(1) Coefficient of variation of demand (CVD)

(2) Ratio of setup costs to inventory holding costs (s/h)

(3) Time between order (TBO)

(4) Length of planning horizon

(5) Lot-sizing method

As the purpose of this experiment is to study the performance of the Wagner-Whitin algorithm, the implementation factor which accounts for different implementations into the algorithm is used in lieu of the lot-sizing method factor. It can also be shown that the ratio of setup costs to inventory holding costs (s/h) is not a relevant experimental factor by the following theorem.

Theorem

Given the first problem and its optimal solution:

(1) demand schedule = $\{ D_1, \ldots, D_n \}$,

(2) ratio of setup costs to inventory holding costs = s/h,

(3) optimal order policy = $\{ Q_1^*, \ldots, Q_n^* \}$.

Then for the following second problem:

(1) demand schedule = $\{ kD_1, \ldots, kD_n \}$,

(2) ratio of setup costs to holding costs

$$= s'/h' = k\,(s/h),\ k=0,$$

the optimal policy is given by $\{ kQ_1^*, \ldots, kQ_n^* \}$

Proof

For the first problem, the total costs is given by

$$\sum_{t=1}^{n} (sY_t + hI_t)$$

where $I_{t-1} + Q_t - D_t = I_t$

$$Y_t = \begin{cases} 1 & , \quad Q_t > 0 \\ 0 & \quad \text{otherwise} \end{cases}$$

$$I_t,\ Q_t \geq 0$$

If $\{Q_1^*, \ldots, Q_n^*\}$ is the optimal inventory policy for this problem, then

$$\sum_{t=1}^{n} \{sY_t^* + hI_t^*\} \le \sum_{t=1}^{n} \{sY_t + hI_t\}$$

where $I_{t-1}^* + Q_t^* - D_t = I_t^*$

$$Y_t = \begin{cases} 1 & , \quad Q_t^* > 0, \\ 0 & , \quad \text{otherwise.} \end{cases} \tag{4.1}$$

Now, if we multiply both sides of (4.1) by $k(h'/h)$, we will have

$$\sum_{t=1}^{n} \{(kh's/h)Y_t^* + (kh')I_t^*\} \le \sum_{t=1}^{n} \{(kh's/s)Y_t + (kh')I_t\}$$

i.e. $$\sum_{t=1}^{n} \{s'Y_t^* + h'(kI_t^*)\} \le \sum_{t=1}^{n} \{s'Y_t + h'(kI_t)\} \tag{4.2}$$

since $(kh's)/h = h'(ks/h)$

$= h'(s'/h')$

$= s'$

Without loss of generosity, we may denote kI_t by I_t' and Y_t by Y_t'. Note I_t' and Y_t' are given by any arbitrary inventory policy $\{kQ_1, \ldots, kQ_n\}$. Inequality (4.2) becomes

$$\sum_{t=1}^{n} \{s'Y_t^* + h'(kI_t^*)\} \le \sum_{t=1}^{n} \{s'Y_t' + h'I_t'\} \tag{4.3}$$

Inequality (4.3) shows that the optimal solution to the second problem is such that

$(Y_t')^* = Y_t^*$

and $(I_t')^* = kI_t^*$. $\qquad (4.4)$

For this problem, its optimal inventory policy, inventory level and demand schedule are related by

$$(I'_{t-1})^* + (Q'_t)^* - kD_t = (I'_t)^*$$

$$\therefore \quad kI^*_{t-1} + (Q'_t)^* - kD_t = kI^*_t \quad \text{from (4.4)}$$

ie. $$(Q'_t)^* = k\{I^*_t + D_t - I^*_{t-1}\}$$

However, from (4.1),

$$Q^*_t = I^*_t + D_t - I^*_{t-1}$$

$$\therefore \quad (Q'_t)^* = kQ^*_t$$

Therefore, the optimal order policy for the second problem is given by $\{kQ^*_1, \ldots, kQ^*_n\}$. □

Now consider the following two problems:

(1) Demand = $\{ D_1, \ldots, D_n \}$

$$s_1/h_1 = 10$$

$$CVD_1 = \sigma_D/\overline{D} \quad , \text{ where } \overline{D} = \sum_{i=1}^{n} D_i \;/\; n$$

$$TBO_1 = \sqrt{(2s_1)/(h_1\overline{D})}$$

(2) Demand = $\{ 100\,D_1, \ldots, 100\,D_n\}$

$$s_2/h_2 = 1000 = 100(s_1/h_1)$$

$$\begin{aligned} CVD_2 &= (100\sigma_D)/(100\overline{D}) \\ &= \sigma_D/\overline{D} \\ &= CVD_1 \end{aligned}$$

$$\begin{aligned} TBO_2 &= \sqrt{(2s_2)/(h_2 100\overline{D})} \\ &= \sqrt{(2)(100s_1)/(h_1 100\overline{D})} \\ &= \sqrt{(2s_1)/(h_1\overline{D})} \\ &= TBO_1 \end{aligned}$$

All the factors for these two problems are identical except the costs ratio, s/h, which shows significant difference. However, based on the previous theorem, these two problems are indeed identical with one's solution being an integral multiple of the other's. Therefore, a problem set should not be characterized by the ratio of setup costs to inventory holding costs (s/h). The value of s/h alone is not meaningful unless it is divided by the mean demand (ie. $s/h\bar{D}$), which is then equivalent to TBO. In conclusion, the use of s/h as an experimental factor is ineffective (as it is included in TBO) and may even lead to incorrect interpretation.

In this experiment, four factors are used, namely, implementation, n, TBO and CVD.

4.2.1 Coefficient of Variation of Demand (CVD)

This factor was first introduced by Kaimann [18] and has then been widely used by many researchers [6, 9, 15, 22]. It measures the difference, adjusted by the mean demand, between demand from period to period and is also termed the lumpiness of demand. All the test results indicate that CVD is an important factor affecting the performance of heuristic rules.

CVD is also expected to affect the computational efficiency of the Wagner-Whitin algorithm when different implementations of the network constraints are incorporated. For example, consider the first implementation, WW1, the lower bound, l_t^*, is given by

$$l_t^* = \max (l_{t-1}^*, l_t)$$

where $l_0^* = 0$

$$l_t = \mathrm{Int}\,\{t - s/(hD_t)\}$$

The lower bound, l_t, will increase as D_t increases. The computational time required for a problem will be greatly reduced when there are some very large D_t's. For example, when there are some t's such that $l_t = t$, the problem will actually be partitioned into several small problems.

The values of CVD used in this experiment are 0.2, 0.6, 1.0 and 1.4, which constitute a range comparable to those used in literature. For example, Berry [6]: 0 to 1.41, Gaither [15]: 0 to 1.41 and Bokko and Whybark [9]: 0.29 to 1.14. It is noted that the data set with a value of 3.31 for CVD used in Berry [6] and Gaither's [15] studies is a special case in which there is only one nonzero demand over the entire planning horizon. In this case, performance of all the lot-sizing rules is identical. Therefore, it is not considered as a legitimate data set for comparison.

4.2.2 Time Between Order (TBO)

Berry [6] first advocated the use of TBO as an alternative factor to s/h used by Kaimann [18]. Since then, most studies have included either one or both of these two factors. It measures conceptually the average number of periods covered by each replenishments. It is found to be a significant factor for the performance of heuristic rules.

TBO is also expected to have impact on the computational efficiencies of different implementations of the network constraints. Again, consider WW1,

$$l_t = \mathrm{Int}\{\ t - s/(hD_t)\ \}$$

As TBO value increases, $s/(hD_t)$ will also increase and hence l_t will decrease. Therefore, the computational time is expected to increase with TBO. The values of TBO used in this experiment as 1, 3, 5, 7, and 9. It is considered as a relatively large range. For example, Berry [6]: 0.73 to 1.82 and Gaither [15]: 1 to 7.

4.2.3 Length of Planning Horizon (n)

The length of planning horizon is a measure of the size of a problem, and hence will vary directly with the computational time. The length of planning horizon has been found to be an important factor affecting the computational efficiencies of heuristic rules. The results in [15] indicated that computational time increases rather linearly with the length of planning horizon for several heuristic rules.

The length of planning horizon is also an important factor which determines the computational time required by the Wagner-Whitin algorithm. In this algorithm, there are n(n+1)/2 entries in the dynamic programming table. Therefore, the computational time is theoretically a polynomial of at least second degree in n. Based on this consideration, computations required by the Wagner-Whitin algorithm will be substantially more than those required by heuristic rules when n becomes very large.

The following values for n are used in this experiment : 24, 48, 72, 96, 120. This is considered to be an adequate range which covers most practical problems. Anderson [3] reported that the average length of planning horizon for inventory policies for material production schedule is only forty weeks.

4.2.4 Implementation Factor

Based on the network constraints, a set of lower bound values can be calculated for the Wagner-Whitin algorithm (see section 3.5 for detailed derivation). When more constraints are used, a larger lower bound can be calculated and hence less computations will be required in the dynamic program. However, it also requires more effort in searching for such a lower bound.

For an efficient implementation of the network constraints, extra searching time for a larger lower bound must be more than compensated by the additional saving in computational time in the dynamic program. In this experiment, the implementation factor accounts for different implementations of the network constraints into the Wagner-Whitin algorithm, as well as the original Wagner-Whitin algorithm. The following notations are used:

Notation	Explanation	Value of a
WW	Original Wagner-Whitin algorithm	-
WW1	One network constraint incorporated	0
WW2	Two network constraints incorporated	1
WW3	Three network constraints incorporated	2
WWT1	All network constraints incorporated	$t-l_t-2$

Note : The value a specifies the range of p values used in equation (3.10) which calculates the lower bound for the dynamic program.

4.3 Demand Data Set Generation

The three factors, n, TBO and CVD used in this experiment are either wholly or partially affected by the demand set pattern. Therefore, it is necessary to develop a procedure which efficiently and effectively generates demand data sets with given values for the experimental factors.

The generation of demand data set with stipulated length of planning horizon (n) is strict forward. For time between order (TBO), which is given by

$$TBO = (2s)/(h\bar{D}),$$

the mean values of all the demand data sets are set at 100. By varying the ratio of setup costs to inventory holding costs (s/h), different TBO values can be obtained.

To generate demand data sets with predetermined values for CVD, the procedure is more complicated. Data sets with sufficiently large range of CVD values (0.2 to 1.4 in this experiment) must be generated. Moreover, the demand in each period must be non-negative.

Most common distributions do not satisfy either one or both of the above two criteria, namely, large CVD values and non-negative demand. Nevertheless, several procedures have been devised for generating demand data sets which fulfill these requirements. For example, McLaren [21], Wemmerlov [27] and Blackburn and Millen [8] have developed such procedures for lot-sizing studies. McLaren's procedure makes use of the uniform distribution. For large CVD values, a compound distribution is used. Wemmerlov's method is based on sampling from a truncated normal distribution. Blackburn and Millen's procedure is again based on a compound distribution, which includes a truncated normal distribution.

In this experiment, demand data sets are generated from a compound distribution derived from two uniform distributions. Our choice is primarily based on the ease of this procedure in eliminating negative demand and its flexibility in handling a large range of CVD values.

For a uniform distribution, x, in the range [a,b],

$$x = (a+b)/2$$

$$\sigma_x = (b-a)/\sqrt{12}$$

$$\therefore\ a = \bar{x} - \sqrt{3}\sigma_x$$

$$b = \bar{x} + \sqrt{3}\sigma_x$$

Now, $x \geq 0 \Rightarrow a \geq 0$

ie. $\bar{x} - \sqrt{3}\sigma_x \geq 0$

$$\sigma_x/\bar{x} \leq 1/\sqrt{3}$$

$$= 0.577$$

$$\therefore\ \quad CVD \leq 0.577$$

Therefore, the range of CVD values is from 0 to 0.577 for a non-negative uniform distribution.

Now, consider two uniform distributions, x_1 and x_2, with means $\bar{x_1}$ and $\bar{x_2}$, and standard deviations σ_{x_1} and σ_{x_2} respectively. Without loss of generosity, let

$\bar{x_2} = a\bar{x_1}$ where $a \geq 1$,

and $\sigma_{x_2} = b\sigma_{x_1}$ where $b = 0$.

Next, consider a compound distribution, x, which is formed by taking r times its total number of demand data from x_1 and the rest from x_2 $(0 \leq r \leq 1)$. Then

$$x_1 = \bar{x}/\{r + a(1-r)\}$$

$$\sigma_{x_1} = \{\sigma_x - r(1-r)(a-1)^2\bar{x}^2/[r + a(1-r)]^2\}^{½}/[r+b^2(1-r)]^{½}$$

Now, $\sigma_{x_1} > 0$ (and $\sigma_{x_2} > 0$)

$$\Rightarrow\ CVD_x > (a-1)\sqrt{r(1-r)}/[r + a(1-r)] \qquad (4.5)$$

For non-negative demand, the lower limits of both x_1 and x_2 must also be non-negative. Therefore,

$$x_1 - \sqrt{3}\sigma_{x_1} \geq 0$$

$$\Rightarrow \quad CVD_x^2 \leq [r + b^2(1-r)]/[\sqrt{3}r + \sqrt{3}a(1-r)]^2 + r(1-r)(a-1)^2/[r + a(1-r)]^2 \qquad (4.6)$$

$$x_2 - \sqrt{3}\sigma_{x_2} \geq 0$$

$$\Rightarrow \quad CVD_x^2 \leq (a/b)^2[r + b^2(1-r)]/[\sqrt{3}r + \sqrt{3}a(1-r)]^2 + r(1-r)(a-1)^2/[r + a(1-r)]^2 \qquad (4.7)$$

Inequalities (4.5) to (4.7) set upper limit for CVD of the compound distribution. By varying r, a and b, a large number of possible ranges for CVD values can be obtained. For example,

r	a	b	Range of CVD values
1.0	any	any	0.00 to 0.58
0.4	5	5	0.58 to 0.88
0.8	5	10	0.88 to 1.15
0.6	15	20	1.04 to 1.33
0.7	20	20	1.30 to 1.61
0.8	20	20	1.58 to 1.92
0.9	15	15	1.75 to 2.10
0.9	20	20	1.96 to 2.34

Using this procedure, it is possible to generate demand data set with CVD value ranges from 0 to 2.34. However, when the number of periods, n, is small, it becomes inefficient to generate a demand set using large r since only a few demand values can be sampled from one of the two uniform distributions with prescribed values for its mean and standard deviation.

With five levels of n and four levels of CVD, twenty demand data sets are generated. Details of these data sets are given in the Appendix. As there are five levels of TBO, there are altogether one hundred problems in this experiment.

4.4 Experimental Results

Computational times taken by the Wagner-Whitin algorithm with different implementations of the network constraints to solve the simulated problems are tabulated in the Appendix. Tables 4.1 to 4.3 show the average computational time for different values of n, TBO and CVD, while tables 4.4 to 4.6 show the average computational times expressed as percentages of those taken by the original algorithm. The relationships between average computational time and each of n, TBO and CVD are depicted in figures 4.1 to 4.6.

4.5 Analysis of Variance

To compare the performance of different implementations of the network constraints and to assess the impact of experimental factors on their performance, an analysis of variance is performed. Such an analysis will provide more rigorous theoretical grounds for our interpretation.

The analysis of variance consists of two stages. In

Table 4.1

Average Computational Times for Different Lengths of Planning Horizon (n)

n	WW	WW1	WW2	WW3	WWT1
24	0.01473	0.00856	0.00720	0.00619	0.00649
48	0.09577	0.02523	0.01766	0.01474	0.01401
72	0.30902	0.04193	0.02931	0.02380	0.02145
96	0.79234	0.06293	0.04119	0.03340	0.03097
120	1.53683	0.07912	0.04902	0.04006	0.03793

Note : All times are given in seconds.

Table 4.2

Average Computational Times for Different Time Between Order Values (TBO)

TBO	WW	WW1	WW2	WW3	WWT1
1	0.00485	0.00404	0.00410	0.00417	0.00414
3	0.61780	0.01142	0.00986	0.01023	0.01056
5	0.70756	0.02779	0.02147	0.01868	0.01992
7	0.70939	0.06036	0.04050	0.03253	0.03173
9	0.70911	0.11415	0.06845	0.05258	0.04452

Note : All times are given in seconds.

Table 4.3

Average Computational Times for Different Coefficients of Variation of Demand (CVD)

CVD	WW	WW1	WW2	WW3	WWT1
0.2	0.56186	0.07259	0.03457	0.02483	0.02146
0.6	0.56351	0.04270	0.02882	0.02341	0.02170
1.0	0.57316	0.03057	0.02551	0.02262	0.02160
1.4	0.50044	0.02836	0.02660	0.02370	0.02392

Note : All times are given in seconds.

Table 4.4

Computational Times of WW1, WW2, WW3 and WWT1
as Percentages of Those of WW for
Different Lengths of Planning Horizon (n)

n	WW1	WW2	WW3	WWT1
24	58.1%	48.9%	42.0%	44.1%
48	26.3%	18.4%	15.4%	14.6%
72	13.6%	9.5%	7.7%	6.9%
96	7.9%	5.2%	4.2%	3.9%
120	5.1%	3.2%	2.6%	2.5%

Table 4.5

Computational Times of WW1, WW2, WW3 and WWT1
as Percentages of Those of WW for
Different Time Between Order Values (TBO)

TBO	WW1	WW2	WW3	WWT1
1	83.3%	84.5%	85.9%	85.2%
3	1.8%	1.6%	1.7%	1.7%
5	3.9%	3.0%	2.6%	2.8%
7	8.5%	5.7%	4.6%	4.5%
9	16.1%	9.7%	7.4%	6.3%

Table 4.6

Computational Times of WW1, WW2, WW3 and WWT1
as Percentages of Those of WW for
Different Coefficients of Variation of Demand (CVD)

CVD	WW1	WW2	WW3	WWT1
0.2	12.9%	6.2%	4.4%	3.8%
0.6	7.6%	5.1%	4.2%	3.9%
1.0	5.3%	4.5%	3.9%	3.8%
1.4	5.7%	5.3%	4.7%	4.8%

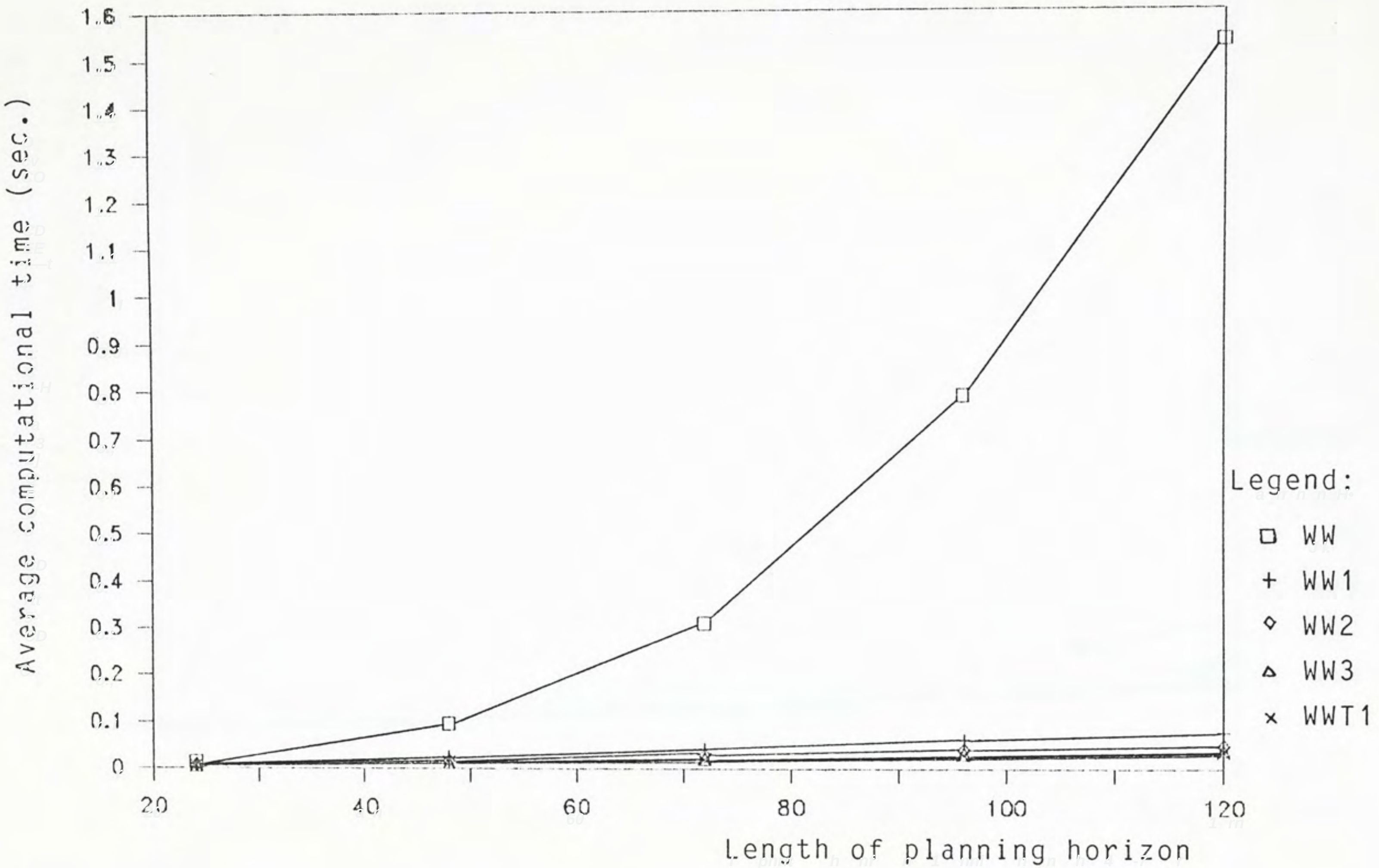


Figure 4.1 Average Computational Time versus Length of Planning Horizon

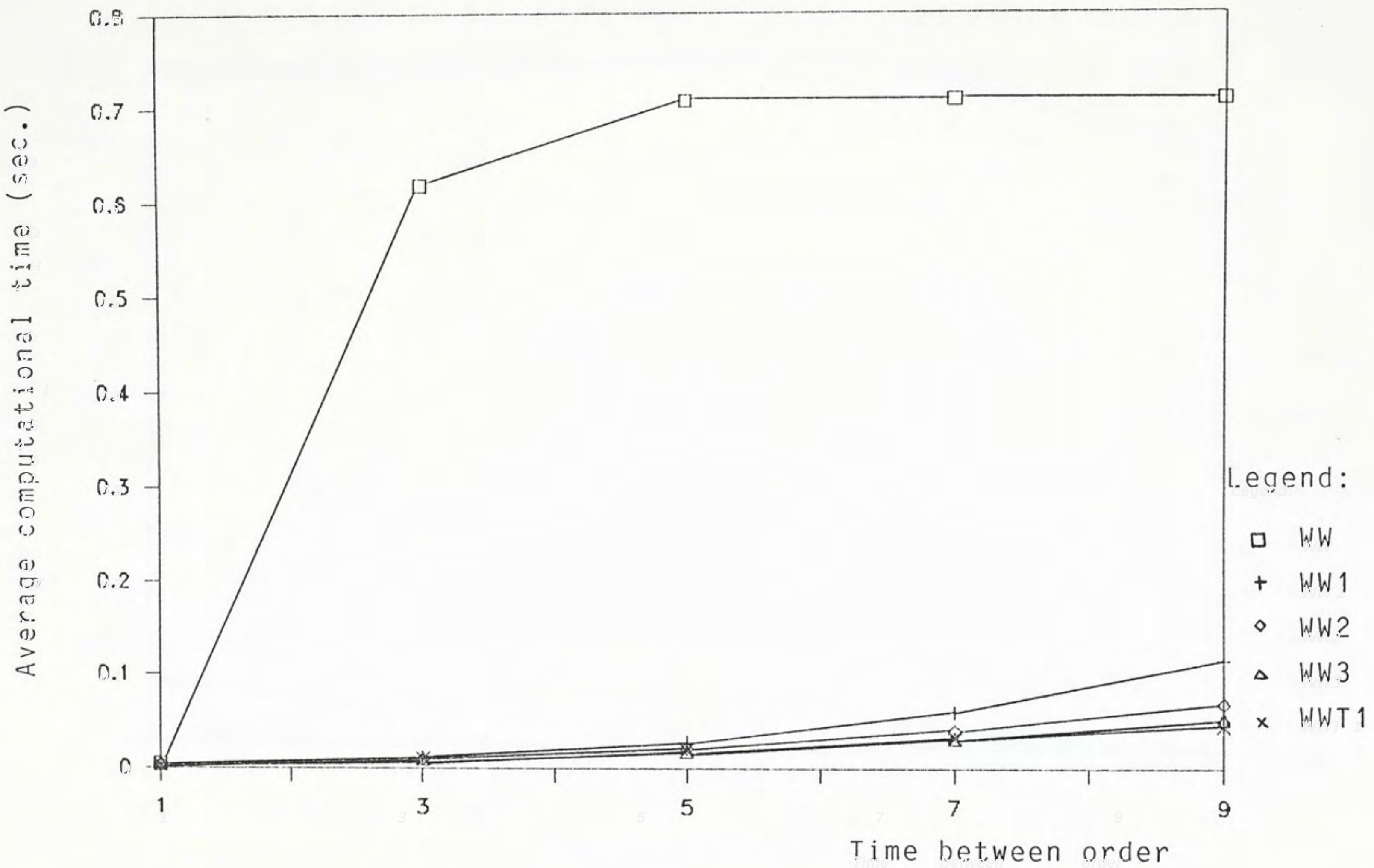


Figure 4.2 Average Computational Time versus Time Between Order

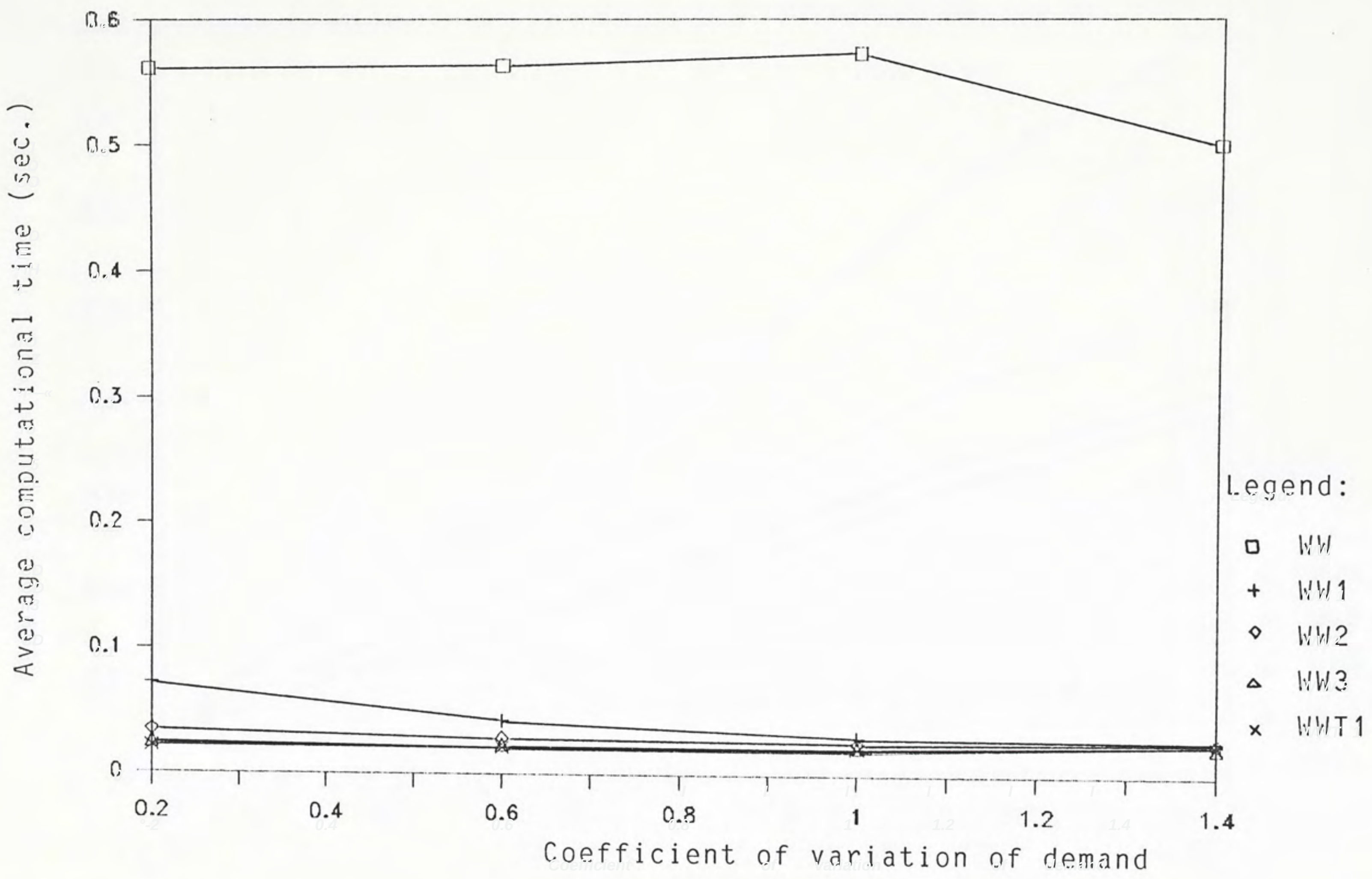


Figure 4.3 Average Computational Time versus Coefficient of Variation of Demand

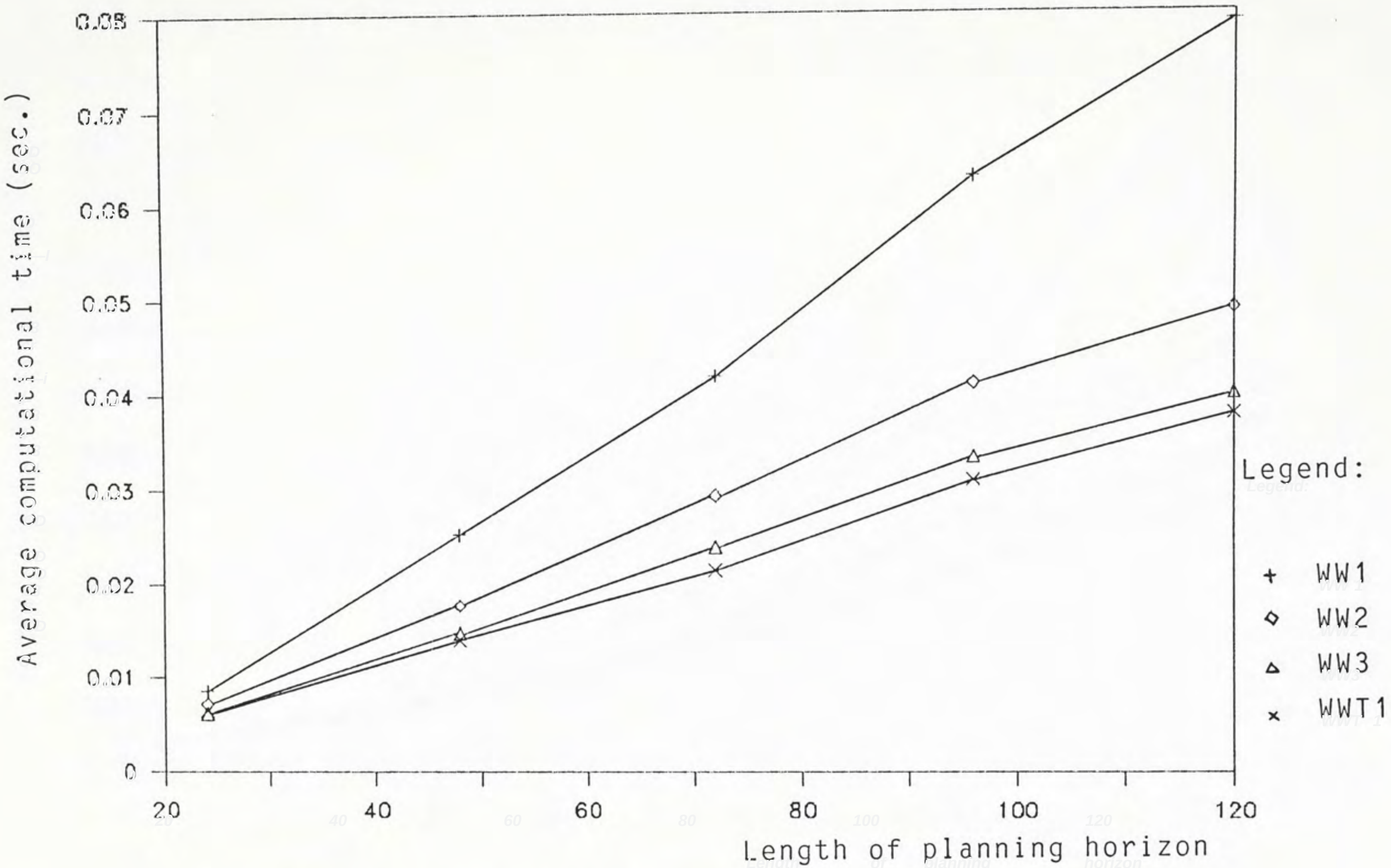


Figure 4.4 Average Computational Time versus Length of Planning Horizon (without WW)

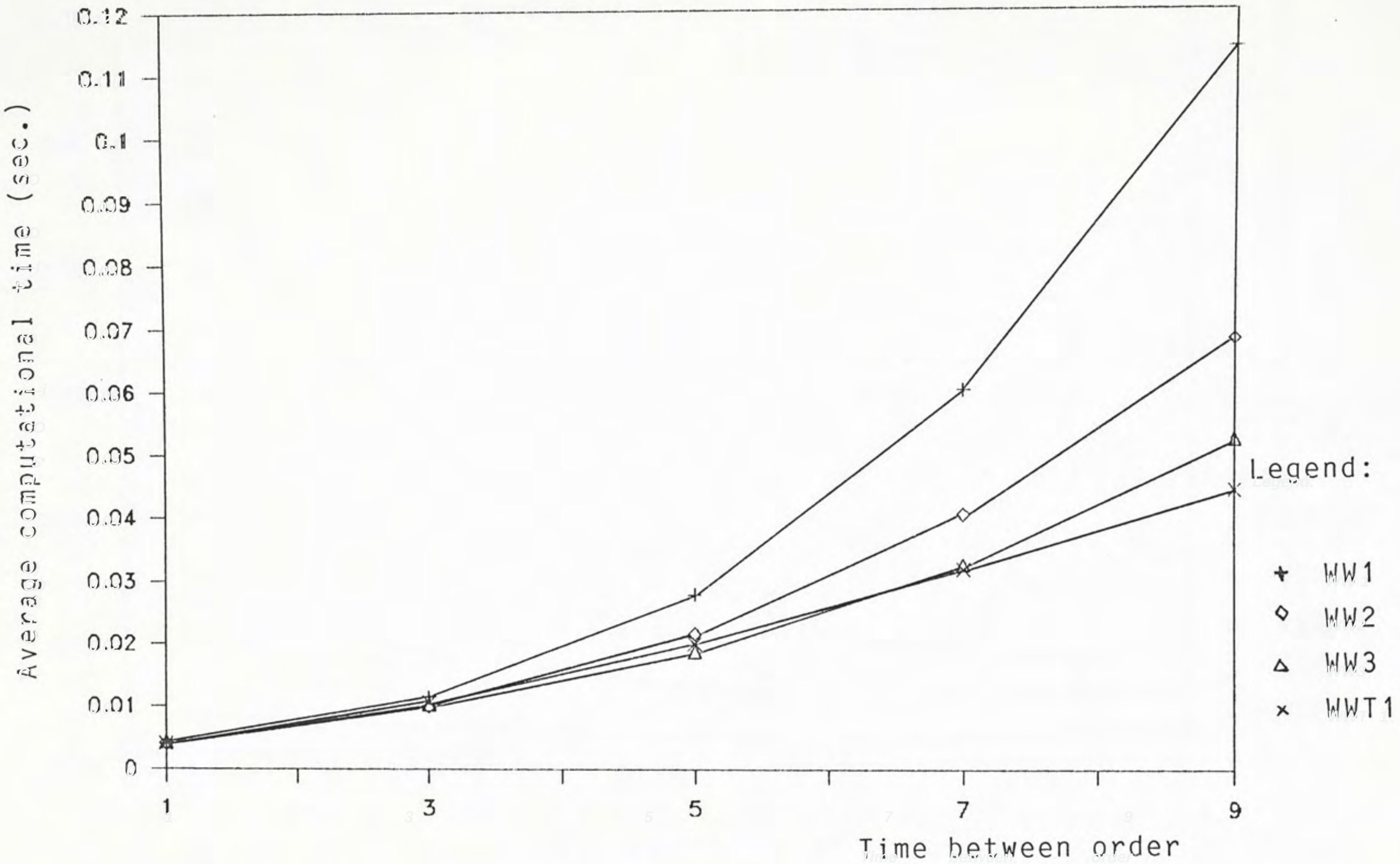


Figure 4.5 Average Computational Time versus Time Between Order (without WW)

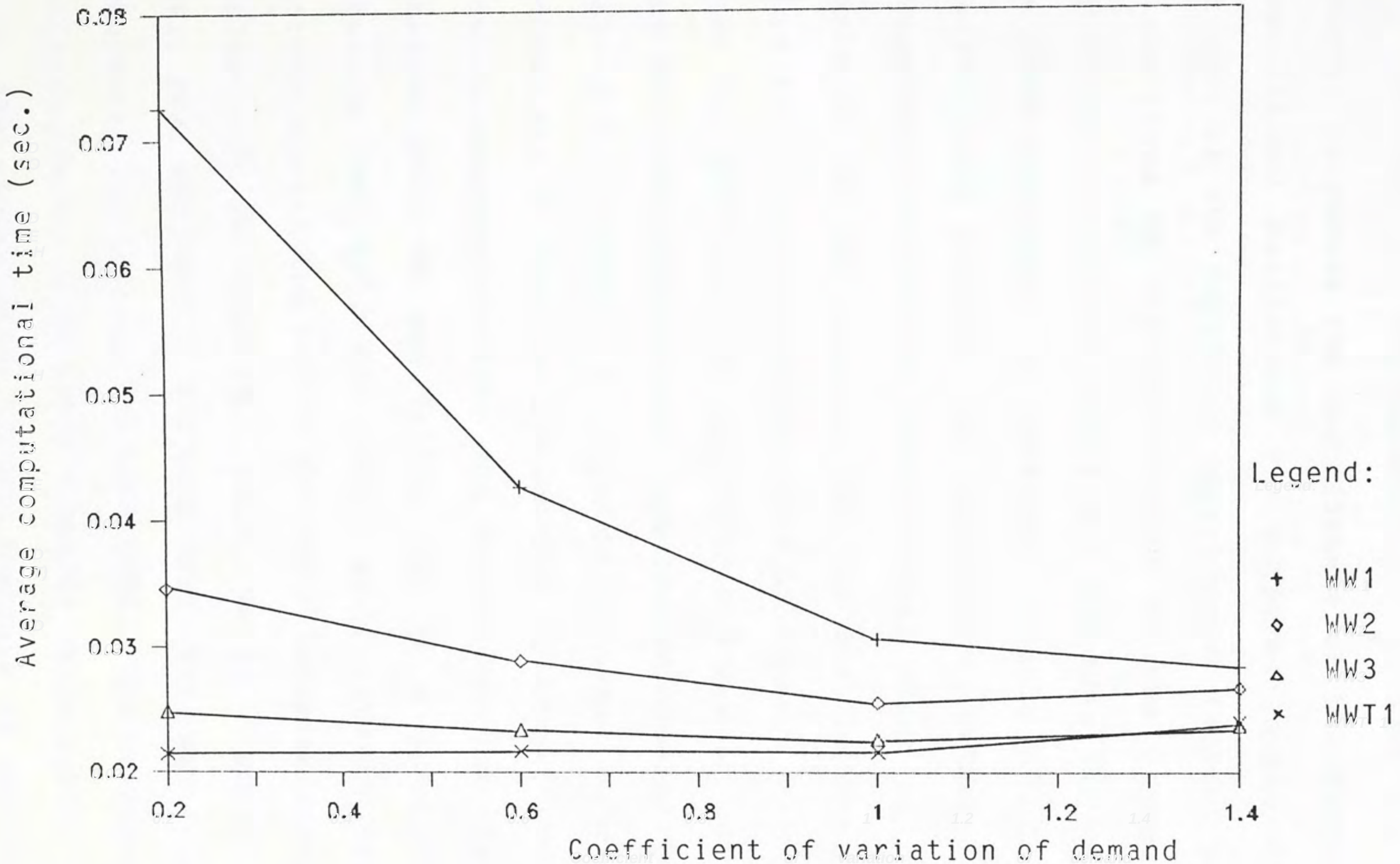


Figure 4.6 Average Computational Time versus Coefficient of Variation of Demand (without WW)

the first stage, a complete model which includes all the four factors (ie. implementation, n, TBO and CVD) is analysed to assess the significance of the difference in computational performance for different implementations, and that of the impact of experimental factors and their interactions on the performance of the Wagner-Whitin algorithm as a whole. Table 4.7 displays the results of the ANOVA analysis. As indicated in table 4.7, CVD is not a significant factor. To investigate further, pairwise comparisons of average computational times at different levels of CVD are computed and the results are shown in table 4.8. Pairwise comparisons of average computational times for different implementations are also computed to test for their significance and the results are given in table 4.9. Finally, a stepwise contrast procedure is carried out to examine the benefit of incorporating more network constraints into the Wagner-Whitin algorithm. Starting with WW and taking one at a time from the sequence {WW, WW1, WW2, WW3}, each implementation is contrasted with the set of all other implementations which employ more constraints. Thus, WW is contrasted with {WW1, WW2, WW3, WWT1}, WW1 with {WW2, WW3, WWT1}, WW2 with {WW3,WWT1} and lastly, WW3 with WWT1. The computations of contrast are given in table 4.10a to table 4.10l.

In the second stage, ANOVA analysis is performed for each implementation to study specifically the impact of experimental factors on individual implementation. The results of ANOVA analysis are shown in tables 4.11a to

4.11e. The factor CVD, which is not significant in the complete model, is found to be significant to some implementations. Therefore, pairwise comparisons of average computational times at different levels of CVD are computed and the results are tabulated in tables 4.12a to 4.12e. To determine the nature of relationships between the computational time and the experimental factors, polynomial contrasts are then computed and the results are given in table 4.13a to table 4.13e.

4.6 Factors Significant to the Performance of the Wagner-Whitin Algorithm

Based on the complete ANOVA model (see table 4.7), the implementation factor is found to be most significant, indicating that different implementations of the network constraints may have significant impact on the computational performance of the Wagner-Whitin algorithm. Detailed comparison of the implementations is presented in the next section. The complete model also reveals that the length of planning horizon (n) and the time between order (TBO) factor are highly significant to all implementations. The coefficient of variation of demand (CVD) is found to be not significant in the complete model, as is confirmed in the pairwise comparisons of means test (See table 4.8). The length of planning horizon (n) measures the size of the problem and therefore the computational time required to solve a problem must be

Table 4.7

ANOVA Table for the Complete Model

Source	F	Level of Significance
Main effects		
Implementation	287.22	0.0000
n	102.88	0.0000
TBO	30.75	0.0000
CVD	0.11	0.9512
Interactions		
Implementation x n	79.57	0.0000
Implementation x TBO	18.20	0.0000
Implementation x CVD	0.46	0.9382
n x TBO	6.67	0.0000
n x CVD	0.12	0.9999
TBO x CVD	0.46	0.9357

Table 4.8

Pairwise Comparisons of Means at Different Levels of CVD for the Complete Model

CVD	Mean	Homogeneous Group s=0.05	Homogeneous Group s=0.10
0.2	0.1365	I	I
0.6	0.1359	I	I
1.0	0.1324	I	I
1.4	0.1277	I	I

Note : 1. s = level of significance.
2. I stands for a homogeneous group within which the means are not significantly different from one another at the specified level of significance.

Table 4.9

Pairwise Comparisons of Means of Different Implementations for the Complete Model

Implementation	Mean	Homogeneous Group s=0.05	Homogeneous Group s=0.10
WW	0.5497	I	I
WW1	0.0435	II	II
WW2	0.0288	II	II
WW3	0.0236	II	II
WWT1	0.0222	II	II

Note : 1. s = level of significance.
2. I and II stand for different homogeneous groups, where a homogeneous group is one within which the means are not significantly different from one another at the specified level of significance.

Table 4.10a

ANOVA Contrasts of WW with {WW1, WW2, WW3, WWT1} for Different Levels of n

n	t-statistic	Level of Significance
24	-0.22	0.8085
48	-2.27	0.0226
72	-8.15	0.0000
96	-21.60	0.0000
120	-43.25	0.0000

Table 4.10b

ANOVA Contrasts of WW1 with {WW2, WW3, WWT1}
for Different Levels of n

n	t-statistic	Level of Significance
24	-0.05	0.9107
48	-0.28	0.7743
72	-0.48	0.6359
96	-0.78	0.4403
120	-1.04	0.3009

Table 4.10c

ANOVA Contrasts of WW2 with {WW3, WWT1}
for Different Levels of n

n	t-statistic	Level of Significance
24	0.02	0.9305
48	0.09	0.8911
72	0.18	0.8363
96	0.24	0.7973
120	0.27	0.7799

Table 4.10d

ANOVA Contrasts of WW3 with WWT1
for Different Levels of n

n	t-statistic	Level of Significance
24	-0.01	1.0000
48	0.02	1.0000
72	0.05	0.9109
96	0.06	0.9098
120	0.05	0.9141

Table 4.10e

ANOVA Contrasts of WW with {WW1, WW2, WW3, WWT1}
for Different Levels of TBO

TBO	t-statistic	Level of Significance
1	-0.02	0.9313
3	-17.68	0.0000
5	-19.97	0.0000
7	-19.46	0.0000
9	-18.61	0.0000

Table 4.10f

ANOVA Contrasts of WW1 with {WW2, WW3, WWT1}
for Different Levels of TBO

TBO	t-statistic	Level of Significance
1	0.00	1.0000
3	-0.03	0.9232
5	-0.22	0.8103
7	-0.72	0.4804
9	-1.66	0.0930

Table 4.10g

ANOVA Contrasts of WW2 with {WW3, WWT1}
for Different Levels of TBO

TBO	t-statistic	Level of Significance
1	0.00	1.0000
3	-0.01	1.0000
5	0.06	0.9088
7	0.22	0.8081
9	0.53	0.6037

Table 4.10h

ANOVA Contrasts of WW3 with WWT1
for Different Levels of TBO

TBO	t-statistic	Level of Significance
1	0.00	1.0000
3	0.01	1.0000
5	0.03	0.9268
7	-0.02	1.0000
9	-0.19	0.8314

Table 4.10i

ANOVA Contrasts of WW with {WW1, WW2, WW3, WWT1}
for Different Levels of CVD

CVD	t-statistic	Level of Significance
0.2	-14.98	0.0000
0.6	-17.68	0.0000
1.0	-17.59	0.0000
1.4	-17.50	0.0000

Table 4.10j

ANOVA Contrasts of WW1 with {WW2, WW3, WWT1}
for Different Levels of CVD

CVD	t-statistic	Level of Significance
0.2	1.09	0.2777
0.6	0.42	0.6788
1.0	0.31	0.7533
1.4	0.54	0.5953

Table 4.10k

ANOVA Contrasts of WW2 with {WW3, WWT1}
for Different Levels of CVD

CVD	t-statistic	Level of Significance
0.2	0.30	0.7551
0.6	0.17	0.8435
1.0	0.12	0.8705
1.4	0.12	0.8731

Table 4.10l

ANOVA Contrasts of WW3 with WWT1
for Different Levels of CVD

CVD	t-statistic	Level of Significance
0.2	0.08	0.8972
0.6	0.05	0.9158
1.0	0.01	1.0000
1.4	0.02	0.9310

Table 4.11a

ANOVA Table for WW

Source	F	Level of Significance
Main effects		
n	442.90	0.0000
TBO	105.54	0.0000
CVD	1.81	0.1579
Interactions		
n x TBO	28.19	0.0000
n x CVD	0.90	0.5570
TBO x CVD	1.71	0.0935

Table 4.11b

ANOVA Table for WW1

Source	F	Level of Significance
Main effects		
n	23.49	0.0000
TBO	59.46	0.0000
CVD	5.66	0.0021
Interactions		
n x TBO	5.81	0.0000
n x CVD	0.72	0.7227
TBO x CVD	8.39	0.0000

Table 4.11c

ANOVA Table for WW2

Source	F	Level of Significance
Main effects		
n	107.88	0.0000
TBO	255.55	0.0000
CVD	5.63	0.0022
Interactions		
n x TBO	19.08	0.0000
n x CVD	0.62	0.8164
TBO x CVD	7.34	0.0000

Table 4.11d

ANOVA Table for WW3

Source	F	Level of Significance
Main effects		
n	328.13	0.0000
TBO	657.86	0.0000
CVD	2.84	0.0474
Interactions		
n x TBO	48.12	0.0000
n x CVD	0.55	0.8735
TBO x CVD	4.21	0.0002

Table 4.11e

ANOVA Table for WWT1

Source	F	Level of Significance
Main effects		
n	422.60	0.0000
TBO	698.57	0.0000
CVD	0.15	0.9306
Interactions		
n x TBO	50.93	0.0000
n x CVD	0.43	0.9409
TBO x CVD	1.39	0.2034

Table 4.12a

Pairwise Comparisons of Means at Different Levels of CVD for WW

CVD	Mean	Homogeneous Group s=0.05	Homogeneous Group s=0.10
0.2	0.5709	I	II
0.6	0.5663	I	I
1.0	0.5659	I	I
1.4	0.4959	I	I

Note : 1. s = level of significance.
2. I and II stand for different homogeneous group, where a homogeneous group is one within which the means are not significantly different from one another at the specified level of significance.

Table 4.12b

Pairwise Comparisons of Means at Different Levels of CVD for WW1

CVD	Mean	Homogeneous Group	
		s=0.05	s=0.10
0.2	0.0616	I	I
0.6	0.0379	II	II
1.0	0.0335	II	II
1.4	0.0413	II	II

Note : 1. s = level of significance.
2. I and II stand for different homogeneous group, where a homogeneous group is one within which the means are not significantly different from one another at the specified level of significance.

Table 4.12c

Pairwise Comparisons of Means at Different Levels of CVD for WW2

CVD	Mean	Homogeneous Group	
		s=0.05	s=0.10
0.2	0.0339	I	I
0.6	0.0284	II	II
1.0	0.0265	II	II
1.4	0.0267	II	II

Note : 1. s = level of significance.
2. I and II stand for different homogeneous group, where a homogeneous group is one within which the means are not significantly different from one another at the specified level of significance.

Table 4.12d

Pairwise Comparisons of Means at Different Levels of CVD for WW3

CVD	Mean	Homogeneous Group s=0.05	Homogeneous Group s=0.10
0.2	0.0252	I	I
0.6	0.0237	I,II	I,II
1.0	0.0225	II	II
1.4	0.0232	II	II

Note : 1. s = level of significance.
2. I and II stand for different homogeneous group, where a homogeneous group is one within which the means are not significantly different from one another at the specified level of significance.

Table 4.12e

Pairwise Comparisons of Means at Different Levels of CVD for WWT1

CVD	Mean	Homogeneous Group s=0.05	Homogeneous Group s=0.10
0.2	0.0222	I	I
0.6	0.0219	I	I
1.0	0.0223	I	I
1.4	0.0224	I	I

Note : 1. s = level of significance.
2. I stands for a homogeneous group within which the means are not significantly different from another other at the specified level of significance.

Table 13a

Polynomial Contrasts for WW

Degree	n	TBO	CVD
1	0.0000/(1565.59)	0.0000/(251.77)	0.0924/(2.95)
2	0.0000/(203.80)	0.0000/(138.06)	0.1646/(1.99)
3	0.1789/(1.86)	0.0000/(30.38)	0.4875/(0.49)
4	-	0.1689/(1.95)	-

Note : level of significance/(F value)

Table 13b

Polynomial Contrasts for WW1

Degree	n	TBO	CVD
1	0.0000/(93.80)	0.0000/(212.50)	0.0075/(7.80)
2	0.8788/(0.02)	0.0000/(24.90)	0.0041/(9.08)
3	0.7940/(0.07)	0.5107/(0.44)	0.7643/(0.09)

Note : level of significance/(F value)

Table 13c

Polynomial Contrasts for WW2

Degree	n	TBO	CVD
1	0.0000/(429.76)	0.0000/(950.05)	0.0008/(12.85)
2	0.4136/(0.68)	0.0000/(71.79)	0.0518/(3.98)
3	0.3161/(1.03)	0.5564/(0.35)	0.8161/(0.05)
4	-	0.1689/(1.95)	-

Note : level of significance/(F value)

Table 13d

Polynomial Contrasts for WW3

Degree	n	TBO	CVD
1	0.0000/(1308.84)	0.0000/(2489.12)	0.0233/(5.49)
2	0.2565/(1.32)	0.0000/(139.63)	0.1022/(2.78)
3	0.1565/(2.07)	0.1156/(2.57)	0.6101/(0.26)

Note : level of significance/(F value)

Table 13e

Polynomial Contrasts for WWT1

Degree	n	TBO	CVD
1	0.0000/(1686.83)	0.0000/(49.63)	0.7178/(0.13)
2	0.6681/(0.19)	0.0000/(43.59)	0.7022/(0.15)
3	0.2097/(1.62)	0.3187/(1.02)	0.6878/(0.16)

Note : level of significance/(F value)

positively correlated to n. This relationship is tested to be significant for all the implementations (see tables 4.11a to 4.11e). With reference to the polynomial contrast calaulation (see tables 4.13a to 4.13e), this relationship is quadratic in n for WW, the original Wagner-Whitin algorithm, and becomes much more linear when the network constraints are incorporated. In fact, as indicated in tables 13a to 13e, F value for the first degree in n increases progressively as more network constraints are included. This trend of increasing linearity can also be observed in figures 4.1 and 4.4.

The time between order factor indicates the complexity of the problem. In general, it is found to be highly significant for all implementations, including the original Wagner-Whitin algorithm (see tables 4.8a to 4.8e). The latter one is a special case in which the average computational time increases sharply as TBO value increases from one, and the effect of TBO quickly levels off (see figure 4.2). When network constraints are implemented, there is a strong quadratic relationship between the computational time and TBO (see figures 4.2 and 4.5, tables 4.13a to 4.13e).

Although the coefficient of variation of demand is found to be not significant in the complete model, it is tested to be significant to some implementations. This discrepancy can be explained by the averaging-off of CVD 's effects in the complete model. CVD is found to be significant for WW1 and WW2 at a significant level of 1%, significant for WW3 at 5% and not significant for WW and WWT1 (see tables 4.11a to 4.11e). Pairwise comparisons of means (see tables 4.12a to 4.12e) indicate that this difference is significant only when CVD value is very small, and in this case the computational time is much larger. Figure 4.6 also shows that the computational times for WW1 and WW2 vary more critically when CVD value is small.

4.7 Comparison of Implementations

As shown in figures 4.1 to 4.3, average computational times required by the Wagner-Whitin algorithm are reduced when the network constraints are implemented. This reduction becomes more substantial as n and TBO values increase, and is quite stable over the tested range of CVD values. Referring to tables 4.4 to 4.6 and figures 4.4 to 4.6, it can be seen that implemention of more network constraints generally leads to saving in computational time, and this benefit is larger as n increases, as TBO value increases and as CVD value decreases. However, marginal saving in computational time decreases as more network constraints are employed. For example, the marginal saving in shifting from WW1 to WW2 is 9.2%, while that in shifting from WW2 to WW3 is only 6.9% for a n value of 24 in table 4.4.

The complete ANOVA model indicates that implementation factor is highly significant (see table 4.7). A close examination based on pairwise comparisons of means (see table 4.9) reveals, however, that the overall difference is significant only between the original Wagner-Whitin algorithm (the null implementation) and the others, ie, between the cases of with and without implementation of network constraints. Overall difference in adding more constraints is not tested to be statistically significant.

The stepwise constrast procedure described in section 4.5 aims at a closer examination of the algorithmic performance at different levels of each experimental factor. Referring to tables 4.10a to 4.10l, it can be seen that saving in computational time by implementating the network constraint is significant except for very small n and TBO values. It is significant at all levels of CVD (see table 4.10i). Although the benefit of adding more constraints is not tested to be statistically significant, its significance increases steadily as either n or TBO increases. This is consistent with our observations in figures 4.4 to 4.6.

As stated in section 4.6, the degree of non-linearity in n for the Wagner-Whitin algorithm is reduced when more network constraints are implemented, and the benefit of implementating more constraints becomes larger as n increases. This is conceivable since a larger lower bound will eliminate more possibilities in the dynamic program, and is therefore more effective, when n is larger.

As TBO value increases, the benefit of implementing more network constraints also increases. This is due to the fact that the value of the lower bound calculated by equation (3.10) is more sensitive to TBO value for small p value in this equation, and the range of p value increases when more constraints are used. As TBO value increases, the value of the lower bound calculated decreases more if the p value in equation (3.10) is smaller, as illustrated below:

Suppose t = 10, D_t = 10 for all t.

TBO	s/h	Int{$t-s/hD_t$}, p=0	Int{$t-1-s/h(D_{t-1}+D_t)$}, p=1
2	20	8	8
4	80	2	5

The value of Int{$t-s/hD_t$} decreases by 6 and that of Int{$t-1-s/h(D_{t-1}+D_t)$} decreases only by 3, when TBO increases by 2.

In section 4.6, it is noted that CVD value will affect the computational performance of WW1 and WW2. However, its effect is significant only when its value is small (see tables 12a to 12e). For the first two network constraints, which correspond to p values of zero and one in equation (3.10), they will calculate for larger lower bounds when there are some very large demand values. Their effectiveness greatly deteriates as the variation in demand decreases. On the other hand, for the more complicated constraints which correspond to larger p values in equation (3.10), their effectiveness is less sensitive to the variation in demand since more demand values are included in these constraints.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

5.1 Conclusions

In this thesis, a constraint set, known as the network constraints, is derived based on a network representation of the single-item, uncapacitated dynamic lot size problem with no backlogging. This constraint set provides lower bound on the evaluation of the recursive function in the Wagner-Whitin algorithm, a dynamic programming formulation of the problem. A Lower Bound Theorem is developed which further improves the computational efficiency of the algorithm by searching for larger lower bound.

Implementation of the network constraints into the Wagner-Whitin algorithm for the dynamic lot size problem with constant costs is investigated. For such a problem, it is characterized by several factors: the length of planning horizon (n), the time between order (TBO) factor and the coefficient of variation of demand (CVD).

Based on the results of a simulation experiment, the computational efficiency of the Wagner-Whitin algorithm is found to be greatly improved when the network constraints are implemented, and this improvement is tested to be statistically significant. In general, implementation of more constraints will lead to more savings in the computation. The benefit of using more constraints is not found to be statistically significant over the tested range of values of the experimental factors. Nevertheless, its significance increases consistently as the values of n and TBO increase.

Results of the experiment also leads to some interesting findings in the relationships between the computational time required by the Wagner-Whitin algorithm and each of the experimental factors. For the original algorithm, computational time is found to be quadratic in n and is not sensitive to other factors except for very small TBO value. When the network constraints are implemented, this relationship with n becomes much more linear. Moreover, a relationship strongly quadratic in TBO value exists. However, CVD is found to be an insignificant factor except for the cases when only the first two constraints are implemented.

5.2 Recommendations for Further Research

The network constraints and the associated Lower Bound Theorem are applicable to the more general cases of the dynamic lot size problem when the costs are all allowed to vary from period to period over the planning horizon. In this thesis, implementation of these constraints into the Wagner-Whitin algorithm is investigated for only the case of constant costs. It will be desirable to pursue further for the cases when one or more costs elements are time-varying.

As mentioned in section 2.6, several researchers have commented that the Wagner-Whitin algorithm is inefficient in computations and its computational time will far exceed those required by the heuristic rules when the length of planning horizon (n) is large . This is primarily due to the nonlinearity in n for the Wagner-Whitin algorithm. With the implementation of the network constraints, the nonlinearity in n is greatly reduced. It is suggested to compare the improved algorithm with some of the widely accepted heuristic rules under a framework comprising the experimental factors studied in this thesis. To evaluate the merit of a heuristic rule, the tradeoff between computational efforts and the quality of the solution obtained should be considered . It is anticipated that, however, with the advancement of computer technology, the aspect of computational efficiency in this consideration will become less significant, thus favouring the development of efficient optimal algorithms.

APPENDIX

COMPUTATIONAL RESULTS OF THE EXPERIMENT

DATA SET	CVD	n	TBO	Computational Times				
				WW	WW1	WW2	WW3	WWT1
1	0.2	24	1	0.001093	0.001406	0.001145	0.001145	0.001145
1	0.2	24	3	0.018177	0.003645	0.002760	0.003072	0.003750
1	0.2	24	5	0.017968	0.010364	0.006406	0.005260	0.005520
1	0.2	24	7	0.018072	0.017708	0.013697	0.008854	0.008385
1	0.2	24	9	0.017968	0.020468	0.017395	0.013229	0.011770
2	0.6	24	1	0.001302	0.001510	0.001406	0.001510	0.001406
2	0.6	24	3	0.017916	0.003385	0.003645	0.003281	0.003385
2	0.6	24	5	0.017968	0.007604	0.005520	0.005104	0.005468
2	0.6	24	7	0.018020	0.014531	0.010208	0.007656	0.008229
2	0.6	24	9	0.018072	0.019062	0.015312	0.011250	0.011874
3	1	24	1	0.001718	0.001666	0.001666	0.001874	0.003281
3	1	24	3	0.018020	0.003645	0.003020	0.003802	0.004374
3	1	24	5	0.018072	0.006354	0.005572	0.005989	0.006406
3	1	24	7	0.017968	0.010208	0.010468	0.009322	0.009999
3	1	24	9	0.017968	0.015625	0.017031	0.014895	0.014062
4	1.4	24	1	0.001874	0.001927	0.001874	0.002187	0.002239
4	1.4	24	3	0.018124	0.002968	0.003437	0.004010	0.003385
4	1.4	24	5	0.018020	0.004895	0.004947	0.004843	0.005937
4	1.4	24	7	0.018229	0.009062	0.007187	0.006874	0.008958
4	1.4	24	9	0.018124	0.015208	0.011250	0.009687	0.010312

DATA SET	CVD	n	TBO	Computational Times				
				WW	WW1	WW2	WW3	WWT1
5	0.2	48	1	0.001927	0.001979	0.002187	0.001979	0.002552
5	0.2	48	3	0.124947	0.007135	0.005625	0.006770	0.006874
5	0.2	48	5	0.125208	0.023385	0.013333	0.011406	0.013437
5	0.2	48	7	0.125520	0.059791	0.028020	0.023177	0.019635
5	0.2	48	9	0.125260	0.109999	0.059010	0.039322	0.030677
6	0.6	48	1	0.002239	0.002447	0.002395	0.002552	0.003229
6	0.6	48	3	0.124426	0.007083	0.005937	0.005937	0.006510
6	0.6	48	5	0.124895	0.013645	0.013229	0.011406	0.012447
6	0.6	48	7	0.124896	0.037395	0.025729	0.020572	0.020312
6	0.6	48	9	0.124947	0.056614	0.042447	0.033385	0.028697
7	1	48	1	0.003072	0.003281	0.002760	0.002864	0.002864
7	1	48	3	0.118072	0.007291	0.006302	0.006666	0.007343
7	1	48	5	0.117863	0.015781	0.012916	0.011874	0.013906
7	1	48	7	0.118072	0.030624	0.025364	0.020885	0.020729
7	1	48	9	0.118281	0.048802	0.040989	0.032760	0.028645
8	1.4	48	1	0.003802	0.004062	0.003645	0.003749	0.003489
8	1.4	48	3	0.058281	0.006822	0.006666	0.006197	0.006406
8	1.4	48	5	0.124583	0.011874	0.010625	0.010625	0.011197
8	1.4	48	7	0.124635	0.019374	0.017395	0.017083	0.017031
8	1.4	48	9	0.124530	0.032187	0.028645	0.025677	0.024218
9	0.2	72	1	0.002708	0.002760	0.002760	0.002708	0.003020
9	0.2	72	3	0.403646	0.011249	0.008958	0.008645	0.009062
9	0.2	72	5	0.406198	0.038436	0.025989	0.016770	0.019010
9	0.2	72	7	0.405364	0.101926	0.050416	0.032760	0.030937
9	0.2	72	9	0.405521	0.199114	0.098802	0.062656	0.044739
10	0.6	72	1	0.003334	0.003698	0.003229	0.003489	0.003333
10	0.6	72	3	0.403124	0.011301	0.009061	0.009583	0.010156

DATA SET	CVD	n	TBO	Computational Times				
				WW	WW1	WW2	WW3	WWT1
10	0.6	72	5	0.402759	0.028281	0.021615	0.023697	0.020364
10	0.6	72	7	0.403020	0.058749	0.053072	0.032604	0.030156
10	0.6	72	9	0.402864	0.111927	0.063489	0.056614	0.042239
11	1	72	1	0.004635	0.004114	0.004843	0.005050	0.003854
11	1	72	3	0.401249	0.009999	0.011354	0.011458	0.009791
11	1	72	5	0.402291	0.019374	0.016979	0.018073	0.017656
11	1	72	7	0.402187	0.038958	0.032447	0.030156	0.027967
11	1	72	9	0.402238	0.069478	0.059166	0.043853	0.041405
12	1.4	72	1	0.006614	0.005052	0.005937	0.005573	0.007344
12	1.4	72	3	0.118228	0.011614	0.011718	0.011666	0.011613
12	1.4	72	5	0.401719	0.021614	0.024687	0.019375	0.019896
12	1.4	72	7	0.401510	0.034999	0.032187	0.029791	0.035417
12	1.4	72	9	0.401354	0.055885	0.049530	0.046509	0.040937
13	0.2	96	1	0.003437	0.003749	0.004010	0.003489	0.003645
13	0.2	96	3	0.939427	0.014531	0.010677	0.012448	0.012916
13	0.2	96	5	0.939426	0.053176	0.028020	0.024271	0.025625
13	0.2	96	7	0.948593	0.141562	0.061875	0.050624	0.041978
13	0.2	96	9	0.940415	0.306249	0.125573	0.084062	0.060937
14	0.6	96	1	0.004270	0.004218	0.004323	0.004791	0.004218
14	0.6	96	3	0.936510	0.015364	0.012344	0.012812	0.013176
14	0.6	96	5	0.935780	0.041041	0.027187	0.025468	0.027656
14	0.6	96	7	0.973489	0.089478	0.060417	0.047395	0.046093
14	0.6	96	9	1.036703	0.167344	0.105781	0.077708	0.062551
15	1	96	1	0.005935	0.005416	0.005260	0.006093	0.005104
15	1	96	3	1.030105	0.012240	0.012447	0.012707	0.012968
15	1	96	5	1.035202	0.026979	0.023697	0.021875	0.023697
15	1	96	7	1.035095	0.055051	0.043645	0.038645	0.039687

DATA SET	CVD	n	TBO	Computational Times				
				WW	WW1	WW2	WW3	WWT1
15	1	96	9	1.033279	0.103541	0.079374	0.062603	0.058280
16	1.4	96	1	0.010253	0.007083	0.009010	0.007656	0.007187
16	1.4	96	3	0.940719	0.018020	0.015208	0.015833	0.016614
16	1.4	96	5	1.032547	0.033854	0.043437	0.030937	0.030885
16	1.4	96	7	1.033737	0.064010	0.068854	0.054165	0.051406
16	1.4	96	9	1.031921	0.095728	0.082655	0.074323	0.074687
17	0.2	120	1	0.005004	0.005573	0.004219	0.004583	0.004739
17	0.2	120	3	2.042648	0.018802	0.013280	0.015156	0.016041
17	0.2	120	5	2.013595	0.067395	0.034114	0.028073	0.031146
17	0.2	120	7	2.012130	0.182083	0.078853	0.055000	0.051562
17	0.2	120	9	2.002136	0.412187	0.167082	0.105313	0.077291
18	0.6	120	1	0.005630	0.005364	0.005311	0.005364	0.005156
18	0.6	120	3	2.022232	0.017135	0.014374	0.014792	0.016249
18	0.6	120	5	2.002395	0.046875	0.031927	0.028541	0.033073
18	0.6	120	7	1.997558	0.100781	0.063177	0.051822	0.052603
18	0.6	120	9	1.983383	0.197708	0.119374	0.087812	0.073905
19	1	120	1	0.008377	0.006771	0.007240	0.007186	0.006197
19	1	120	3	1.980407	0.018750	0.016770	0.016354	0.017291
19	1	120	5	2.026306	0.039113	0.033645	0.032135	0.033957
19	1	120	7	2.022186	0.074114	0.063541	0.058280	0.054166
19	1	120	9	1.990463	0.137084	0.101249	0.085104	0.076458
20	1.4	120	1	0.019851	0.008802	0.008854	0.009531	0.008698
20	1.4	120	3	0.639587	0.027500	0.023541	0.023437	0.023229
20	1.4	120	5	1.988388	0.040833	0.045469	0.037916	0.041041
20	1.4	120	7	1.987503	0.066770	0.063437	0.054843	0.059270
20	1.4	120	9	1.986770	0.108749	0.084895	0.079895	0.076615

Note : All computational times are given in seconds.

BIBLIOGRAPHY

1. Adam, E.E. Jr and Ebert, R.J., Production and Operations Management : Concepts, Models and Behaviour, Prentice-Hall International Editions, third edition, 1986.

2. Aggarwal, S.C., "A Review of Current Inventory Theory and Its Applications," International Journal of Production Research, Volume 12, Number 4, 1974, pp. 443-482.

3. Anderson, J.C., Schroeder, R.G., Tupy, S.E., and White, E.M., "Material Requirements Planning Systems: The State of Art", Production and Inventory Management, Volume 23, 1983, pp. 51-67.

4. Aucamp, D.C. and Fogarty, D.W. , " Lot Sizing in MRP ", APICS 1982 Conference Proceedings, pp. 1-4.

5. Bahl, H.C. and Stanley Zionts, "Lot Sizing as a Fixed-Charge Problem", Production and Operations Management, First Quarter, 1986, pp. 1-10.

6. Berry, W.L., " Lot Sizing Procedures for Requirements Planning Systems : A Framework for Analysis ", Production and Inventory Management, Volume 13, Number 2, June 1972, pp. 19-34.

7. Blackburn, J.D. and H. Kunreuther, "Planning Horizons for the Dynamic Lot Size Model with Backlogging", Management Science, Volume 21, Number 3, November 1974, pp. 251-255.

8. Blackburn, J.D. and R.A. Millen, "Selecting a Lot

Sizing Technique for a Single Level Assembly Process: Part II - Empirical Results", Production and Inventory Management, 1979, 20(5), pp. 41-52.

9. Bobko, P.B. and Whybark, D.C.," The Coefficient of Variation as a Factor in MRP Research", Decision Sciences, Volume 16, 1985, pp. 421-427.

10. Boe, W.J. and Yilmaz, C., "The Incremental Order Quantity", Production and Inventory Management, Volume 24, Number 2, 1983, pp. 94-100.

11. Chase, R.B. and Aquilano N.J., Production and Operations Management: A Life Cycle Approach, Richard D. Irwin, Inc., Illinois, 1973.

12. Chirakiti, Prakob, A Study of Lot Sizing Methods for Material Requirements Planning Systems, unpublished Ph.D. Thesis, University of Missouri-Columbia, 1983.

13. De Matties, J.J., "An Economic Lot Sizing Technique", IBM Systems Journal, Volume 7, 1968, pp. 30-48.

14. Eppen, G.D., F.J. Gould and B.P. Pashingian, "Extensions of the Planning Horizon Theorem in the Dynamic Lot Size Model", Management Science, Volume 15, Number 5, January 1969, pp. 268-277.

15. Gaither Norman, " An Improved Lot-Sizing Model for MRP Systems", Production and Inventory Management, Third Quarter, 1983, pp. 11-19.

16. Groff, G.K., "A Lot-Sizing Rule for Time-Phased Components Demand", Production and Inventory Management, Volume 20, First Quarter, 1979, pp.47-53.

17. Hadley, G. and Whitin, T.M., Analysis of Inventory

Systems, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1963.

18. Kaimann, R.A., "EOQ vs Dynamic Programming - Which One to Use for Inventory Ordering", Production and Inventory Management, 1969, 10(4), pp.66-74.

19. Karni, R., "Maximum Part-Period Gain (MPG) - A Lot-Sizing Procedure for Unconstrained and Constrained Requirements Planning Systems", Production and Inventory Management, Volume 22, Number 2, 1981, pp. 91-98.

20. Law, J.S. and Khumawala, B.M., "An Efficient Heuristic for the Single-Item Discrete, Lot-Sizing Problem", working paper, Faculty of Business Administration, The Chinese University of Hong Kong, 1987.

21. McLaren, B.J., A Study of Multiple Level Lot Size Procedures for Material Requirements Planning Systems, unpublished Ph.D. Dissertation, Purdue University, 1977.

22. Ritchie Eric and Tsado, A.K., "A Review of Lot-Sizing Techniques for Dterministic Time-Varying Demand", Production and Operations Management, Third Quarter, 1986, pp. 65-79.

23. Silver, E.A. and Meal, H.E., "A Heuristic for Selecting Lot-Size Quantities for the Case of a Deterministic Time-Varying Demand Rate and Discrete Opportunities for Replenishment", Production and Inventory Management, Volume 14, Number 2, Second Quarter 1973, pp. 64-75.

24. Wagner, H.M., Principles of Operations Research, Prentice-Hall, Inc., Englewood Cliffs, N.J., second edition, 1975.

25. Wagner, H.M. and Whitin, T.M., " Dynamic Version of the Economic Lot Size Model", Management Science, Volume 5, Number 1, October 1951, pp. 80-96.

26. Wemmerlov, U., "Design Factors in MRP Systems : A Limited Survey ", Production and Inventory Management, Vol. 20, Fourth Quarter, 1979, pp.15-35.

27. Wemmerlov, U., "A Comparison of Discrete, Single State Lot-Sizing Heuristics with Special Emphasis on Rules Based on the Marginal Cost Principle", Engineering Costs and Production Economics, 1982, 7, pp. 45-58.

28. Wemmerlov , U. and Whybark, D.C., "Lot-Sizing Under Uncertainty in a Rolling Schedule Environment", International Journal of Production Research, Volume 22, Number 3, 1984, pp. 467-484.

29. Zabel, E., " Some Generalizations of an Inventory Planning Horizon Theorem", Management Science, Volume 10, Number 3, April 1964, pp. 465-471.

30. Zangwill, W.I., "A Deterministic Multi-Period Production Scheduling Model with Backlogging", Management Science, Volume 13, November 1, September 1966, pp. 105-119.

31. Zangwill, W.I., "Minimum Concave Flows in Certain Networks", Management Science, Volume 14, November 7, March 1968, pp. 429-450.

32. Zangwill, W.I.," A Backlogging Model with a Multi-Echelon Model for a Dynamic Lot Size Production System - A Network Approach", Mangement Science, Volume 15, November 9, May 1969, pp. 506-527.